

附送多媒体光盘(视频讲解+本书源代码)

网站开发非常之旅

珍藏版

强锋科技

张金霞 编著

HTML 网页设计参考手册

- 内容全面，囊括HTML、JavaScript、CSS等各方面的内容
- 贯穿大量的实例和最终执行效果，具有很强的实用性
- 配合Dreamweaver进行讲解，符合当前网页制作的技术特点
- 代码规范，层次清楚，注释丰富，易于理解

清华大学出版社

网站开发非常之旅

HTML网页设计参考手册

强锋科技

张金霞　编著

清华大学出版社

北　京

内 容 简 介

HTML 语言是制作网页的基础语言。作为一个网页制作爱好者或者专业的网页设计师，HTML 语言知识是不可或缺的。本书由最基本的 HTML 语法开始讲解网页设计的基础技术，详细介绍了各种网页制作的标记；然后介绍如何运用 CSS 控制网页画面中文字与图片的样式；接下来讲解了 JavaScript 语言与网页特效的制作；最后以应用最广泛的 Dreamweaver 为例，介绍网页设计的方法。在讲解中配有大量范例，使读者在实际操作中学习制作网页。

本书适合作为各类网页设计制作培训学校的教材，也可供想要把网页做得更好的广大普通网页制作爱好者学习，以及从事网站建设和网页设计制作的专业人士分析借鉴。

图书在版编目（CIP）数据

HTML 网页设计参考手册/张金霞编著. —北京：清华大学出版社，2006. 7
（网站开发非常之旅）

ISBN 7-302-13170-8

I. H…　II. 张…　III. 超文本标记语言，HTML－主页制作－程序设计　IV. TP393.092
中国版本图书馆 CIP 数据核字（2006）第 059680 号

出 版 者：清华大学出版社　　**地　　址：**北京清华大学学研大厦
http://www.tup.com.cn　　**邮　　编：**100084
社 总 机：010-62770175　　**客户服务：**010-62776969
组稿编辑：欧振旭
文稿编辑：李虎斌
封面设计：范华明
版式设计：李付莉
印 刷 者：
装 订 者：
发 行 者：新华书店总店北京发行所
开　　本：203×260　**印张：**23　**字数：**505 千字
版　　次：2006 年 7 月第 1 版　2006 年 7 月第 1 次印刷
书　　号：ISBN 7-302-13170-8/TP・8328
印　　数：1～6000
定　　价：39.00 元（附光盘 1 张）

编 辑 寄 语

“不积跬步，无以至千里”

——中国思想家荀子

2006 年 1 月，北京图书订货会，在清华大学出版社的展位上，众多发行商和书店经营人对刚出版的一套计算机图书表示了极大的关注。这套书就是“网站开发非常之旅”。在琳琅满目的计算机图书中，这套书缘何有这样的“魔力”呢？在和他们的交流中我了解到，当前网络类图书的热卖和我们这套书的精品感觉是他们看好这套书的主要原因。

时隔半年之久，再看看这套书，还真没有辜负众人的期望。在短短的半年内这套书已多次印刷，销售居同类图书前列。对于计算机图书，这是一个让人欣慰的结果。

当然，好卖的书未必一定就是好书。这得由读者说了算。根据半年以来读者的反馈可以看出，他们比较一致地认为这套书定位明确，内容有特色，编写质量较好，看后能学到真正有用的东西。我们从中不难看出这套书受欢迎的一些原因。

为了让读者更加全面地了解这套书，下面具体介绍这套书的定位、内容、特色、读者对象等。

丛书定位

根据读者的实际需求，本丛书并不追求面面俱到，而是**定位于实用，并注重对基本知识点的掌握和对基本技能的提升。**突出体现在以下几点：

- ☑ 每本书大体都对应着相应的工作岗位，着重让读者掌握一项技能，使他们在学完一本书以后，可以将所学应用到实际工作中去。
- ☑ 讲解概念，但并不拘泥于概念，而是侧重于对概念的灵活运用，从而让读者在实践中加深对概念的理解和对基本知识的掌握。
- ☑ 不安排纯演示性实例，那种实例通常没有任何应用价值，读者很难通过它而达到较好的应用水平。本丛书中的实例大多选自于实际开发，是对作者多年开发经验的总结。读者通过对这些实例的研读，可以真正体验实际的开发过程，从而将所学应用于实践。

丛书书目

第一批（已出版）：

《ASP 网络编程从入门到精通》

《ASP.NET 网络数据库开发实例精解》

《ASP+SQL Server 典型网站建设案例》

《服务器配置全攻略》

第二批（2006 年 9 月出版）：

《ASP.NET 2.0 网络编程从入门到精通》
《ASP.NET 2.0+SQL Server 网络应用系统开发案例精解》
《Dreamweaver 网页制作与色彩搭配全攻略》
《JavaScript 网页特效实例大全》
《HTML 网页设计参考手册》

后续出版计划（书名暂定）：

《PHP 网络编程从入门到精通》
《PHP+MySQL 典型网站建设案例精解》
《JSP 网络编程从入门到精通》
《Dreamweaver 网页制作完全手册》
《网页设计的艺术》

丛书特色

☑ 本丛书**按照网站开发的流程组织内容**，从最初的服务器配置，到后台编码的实现，到前台网页设计等内容均有涉及，真正实现了网站开发一条龙。学完本丛书，读者可以全面掌握网站开发的各项技术。
☑ **作者均为有丰富的网络编程经验的一线开发人员**，可以确保技术的先进性、实用性和深入性。
☑ 内容讲解到位，避免空洞，**每个知识点都配有实例**，读者可以上机操作，体会其中的奥妙。
☑ **贯穿丰富的应用实例，并专门设计了综合案例，**让读者真正做到学以致用，并领会项目开发的整体思路。
☑ **配视频讲解光盘，**将一些简单的、操作性强的内容放到光盘中，以加速读者的学习，并节省篇幅，降低定价，让读者真正得到实惠。
☑ **提供必要的售后服务。**丛书的每本书都有相应的技术论坛与作者 E-mail（见前言），读者若有疑问，可以按此寻求帮助。

读者对象

☑ 网站建设及网络开发人员
☑ 网页设计及制作人员
☑ 网站维护人员
☑ 网页制作爱好者
☑ 大中专院校的学生
☑ 毕业设计的学

“不积跬步，无以至千里”。我想这句话可能最能概括这套书成功的真正原因。这套书要想走得远，就得一步一个脚印地做好各个环节的工作。从对读者需求的调研，到确立写作思路，到选择作者，到后期编辑及制作，乃至技术服务等，都得一一落实。只有这样，才能给这套书的继续发展奠定基础，才能让这套书继续成为广大读者的良师益友。

在“网站开发非常之旅”第二批图书即将出版之时，作为这套书的编辑，自是有感而发，希望这套书能一如既往地以高质量、高品质和读者见面，同时也希望这套书能继续受到读者的关注和青睐。

本书编辑
2006 年 9 月于清华园

前　　言

随着 Internet 的迅速发展，网页设计技术已成为学习计算机的重要内容之一。目前大部分制作网页的方式都是运用网页设计软件，这些软件的功能相当强大，使用非常方便。不论是哪一种网页设计软件，最后都是将所设计的网页转化为 HTML、CSS、JavaScript、VBScript 或者 ASP。所以说，虽然有这些网页设计软件帮助我们设计网页，但是一个专业网页设计者仍需了解 HTML、CSS、JavaScript、VBScript 及 ASP 等网页设计语言和技术的使用，这样才能充分发挥自己丰富的想象力，更加随心所欲地设计网页，以实现网页设计软件不能实现的许多重要功能。

上面介绍的几种网页设计语言中，VBScript 和 ASP 注重编程技巧，涉及的知识很多，本书不作讲解。本书集中讲解 HTML、CSS、JavaScript 这 3 种语言的语法要素，几乎涉及了它们的所有内容，读者可以把它作为参考手册，放在案头随时查阅。

本书以最新的 HTML 4 为基础，系统介绍了文字、链接、列表、表格、表单、图像、多媒体、多窗口等方面的元素和属性。

在 CSS 部分，本书完整地介绍了 CSS 的设计思想和关键概念，特别对 CSS 的各种属性和属性值进行了详细的解释和分析。

在 JavaScript 部分，主要介绍了 JavaScript 脚本语言的主要特征和基本功能，讲解了编写 JavaScript 脚本的规则以及如何在 HTML 文档中嵌入 JavaScript 脚本语言等内容，同时在最后还给出了常用的脚本范例，供读者学习。

考虑到实际设计网页时大部分用户都是使用 Dreamweaver 作为设计工具，而不是手工编写脚本，所以本书在开始处给出了 Dreamweaver 的基本用法，在最后一章向读者展示了如何在 Dreamweaver 中实现前面讲解的大部分功能，让读者对网页设计有一个整体的认识。

本书的特点：

- ❑ 内容全面，囊括了 HTML、JavaScript、CSS 各方面的内容，适合作为案头手册随时查用。
- ❑ 真正做到了不讲则已，一讲必通，在讲解各个标记和属性时，全部配有实例和执行效果，让读者领会其用法。
- ❑ 针对当前网页设计的实际情况，配合 Dreamweaver 进行讲解。

本书具有较强的可操作性与实用性，既适用于从事网页设计、制作的专业人士学习 HTML、JavaScript 语言和 CSS 技术，又可作为广大网友学习网页制作的入门教材。

本书由张金霞组织编著，其他参与编写、资料整理和程序调试的人员有王俊标、陈晨、李卓龙、高守传、郭瑞、周宇炜、蔡雪焘、陈杰、荣飞、郑林、张路平、项宇峰、罗皓菡、赵正坤、公芳亮、程明雷、梁文建、马斗、邱哲、宋昕、陈刚等。在此对大家的辛勤工作表示衷心的感谢！

本书的专有论坛：http://www.douban.com/group/19074/。作者的 E-mail：martt0656@163.com。

编著者

2006 年 9 月

目　录

第 1 章

HTML 入门

- HTML 的基本概念
- HTML 的发展历史
- 网页设计原则
- Dreamweaver 简介
- 在 Dreamweaver 中直接编写 HTML

网页制作技术日新月异，但都是以 HTML 为基础的。HTML 是浏览器识别网页的标记语言，可以说，没有它就没有丰富多彩的网页。本章首先让读者对 HTML 有一个初步的认识，了解其概念、发展历史以及编写方法。

1.1　HTML 的基本概念

1.1.1　HTML 简介

HTML 的英文全称是 Hyper Text Markup Language，直译为超文本标记语言。它是全球广域网上描述网页内容和外观的标准。HTML 包含了一对打开和关闭的标记，在当中包含有属性和值。标记描述了每个在网页上的组件，例如文本段落、表格或图像等。

事实上，HTML 是一种因特网上较常见的网页制作标注性语言，而并不能算做一种程序设计语言，因为它缺少程序设计语言所应有的特征。HTML 通过 IE 等浏览器的翻译，将网页中所要呈现的内容、排版展现在用户眼前。

1.1.2　HTML 的结构概念

一个完整的 HTML 文件包括标题、段落、列表、表格以及各种嵌入对象，这些对象统称为 HTML 元素。在 HTML 中使用标签来分割并描述这些元素。实际上可以说，HTML 文件就是由各种 HTML 元素和标签组成的。一个 HTML 文件的基本结构如下：

```
<html>          文件开始标记
<head>          文件头开始的标记
…               文件头的内容
</head>         文件头结束的标记
<body>          文件主体开始的标记
…               文件主体的内容
</body>         文件主体结束的标记
</html>         文件结束标记
```

从上面的代码结构可以看出，在 HTML 文件中，所有的标记都是相对应的，开头标记为< >，结束标记为</ >，在这两个标记中间添加内容。

有了标记作为文件的主干后，HTML 文件中便可添加属性、数值、嵌套结构等各种类型的内容了。

1.1.3　HTML 的标记

既然 HTML 是超文本标记语言，那么可以想象其构成主要是通过各种标记来标示和排列各对象，通常由尖括号“<”、“>”以及其中所包容的标记元素组成。例如，<head>与</head>就是一对标记，称为文件的头部标记，用来记录文档的相关信息。

在 HTML 中，所有的标记都是成对出现的，而结束标记总是在开始标记前增加一个“/”。标记与标记之间还可以嵌套，也可以放置各种属性。此外在源文件中，标记是不区分大小写的，因此在 HTML 源程序中，<Head>与<HEAD>的写法都是正确的，而且其含义是相同的。

HTML 定义了 3 种标记用于描述页面的整体结构。页面结构标记不影响页面的显示效果，它们是

帮助 HTML 工具对 HTML 文件进行解释和过滤的。

- <html>标记：HTML文档的第 1 个标记，它通知客户端该文档是HTML文档，类似地，结束标记</html>出现在HTML文档的尾部。
- <head>标记：出现在文档的起始部分，标明文档的头部信息，一般包括标题和主题信息，其结束标记</head>指明文档标题部分的结束。
- <body>标记：用来指明文档的主体区域，该部分通常包容其他字符串，例如标题、段落、列表等。读者可以把HTML文档的主体区域简单地理解成标题以外的所有部分，其结束标记</body>指明主体区域的结尾。

1.2　HTML 的发展历史

1969 年前后，托德 • 尼尔逊提出超文本的概念，IBM 公司的 Charkes Goklfard 等设计出了通用标记语言——GML。到 1978 年，美国国家标准局一工作组对 GML 进行了规范，推出了命名为 SGML 的通用标记语言。1980 年，ISO 正式确定 SGML 为描述各种电子文件结构及内容的国际通用标准。

1990 年，Tim Berners-Lee 将他设计的初级浏览和编辑系统在网上合二为一，创建了一种快速小型超文本语言来为他的想法服务。他设计了数十种乃至数百种未来使用的超文本格式，并想象智能客户代理通过服务器在网上进行轻松谈判并翻译文件。它同 Macintosh 的 Claris XTND 系统极为相似，不同的是它可以在任何平台和浏览器上运行。

最初的 HTML 语言以文本格式为基础,可以用任何编辑器和文字处理器来为网络创建或转换文本，仅有不多的几个标签。网络从此迅猛发展，人们开始在网上发布信息。很快人们就开始琢磨在网上放置图像和图标。

1993 年，NCSA 推出了 Mosaic，也就是第一个图文浏览器，从此 Web 开始迅速地发展起来。HTML 语言也不断产生新型、功能强大且生动有趣的标签形式，例如<background>、<frame>、<font>和<blink>等。

但是此时，出现了许多不同的 HTML 版本，而只有设计者和用户共有的 HTML 部分才可以正确显示。因此在这段时间，W3C 都在激烈争论名叫 HTML 3 的新技术，该文件概括了所有全新的特性，但没有任何技术支持。出于这种混乱局面的考虑，在 1996 年，W3C 的 HTML Working Group 组织编写了新的规范，从此 HTML 3.2 开始发展，它更接近于现实的目标，即提供给内容商和浏览器发展商在研究工作中一个公允的参考标准。

到现在为止，HTML 已经发展到了比较成熟的 HTML 4 版本，在这个版本下的语言中，规范更加统一，浏览器之间的兼容性也更加完善。

1.3　网页设计原则

在设计网页时，一般要遵循以下原则：

- 结构性：在设计网页时，须注意网页的标记结构、脚本语言结构、使用条列的方式、善用分段及空白字符，让整体结构看起来整齐美观，易于纠错及理解，在发生问题时，才能很快找

到错误所在处。

- 通用性：考虑标记语言能否适用于各种浏览器，尽量以大多数浏览器都支持的标记语言为主，倘若有不得已的情况，一定要特别加以注明，并找出替代性的方案。遇有可以注解或说明的标记或组件属性，应该尽可能地注明其内容。
- 差异性：了解各种浏览器的差异，力求输出的结果尽可能一致。
- 习惯性：了解用户使用窗口与浏览时的习惯，如组件摆放的顺序，习惯用鼠标、Tab键、Esc键及Enter键等操作习惯，或因网页下载时间太长，无法让用户耐心等候等因素，都是设计网页时必须考虑的因素。
- 适用性：有些组件会因用户的窗口环境或安装的程序而异，可能在作者的计算机里可以完整地执行，而在其他用户的计算机上却只能下载文档，所以在设计完毕后，一定要多试几组不同平台的计算机，以确保网页的可行性。
- 反复性：反复检查是否错误，是否有需要注意文字大小写之处，以及名称是否正确。

1.4　Dreamweaver 简介

使用 Dreamweaver，设计师可以随心所欲地编写代码、设计网站网页以及进行高级开发。无论是喜欢手写 HTML 代码还是习惯于可视化环境，Dreamweaver 都能提供方便快捷、功能强大的工具，使用 Dreamweaver 将是一种全新的体验。在易用、创新、规范等优点的基础上，Dreamweaver 还拥有更先进的网页布局和设计环境，以及更为强大的代码编辑功能等卓越特性。

Dreamweaver MX 2004 的操作界面主要由以下几个部分组成：标题栏、菜单栏、插入栏、文档工具栏、文档窗口、属性面板以及多个浮动面板，如图 1-1 所示。

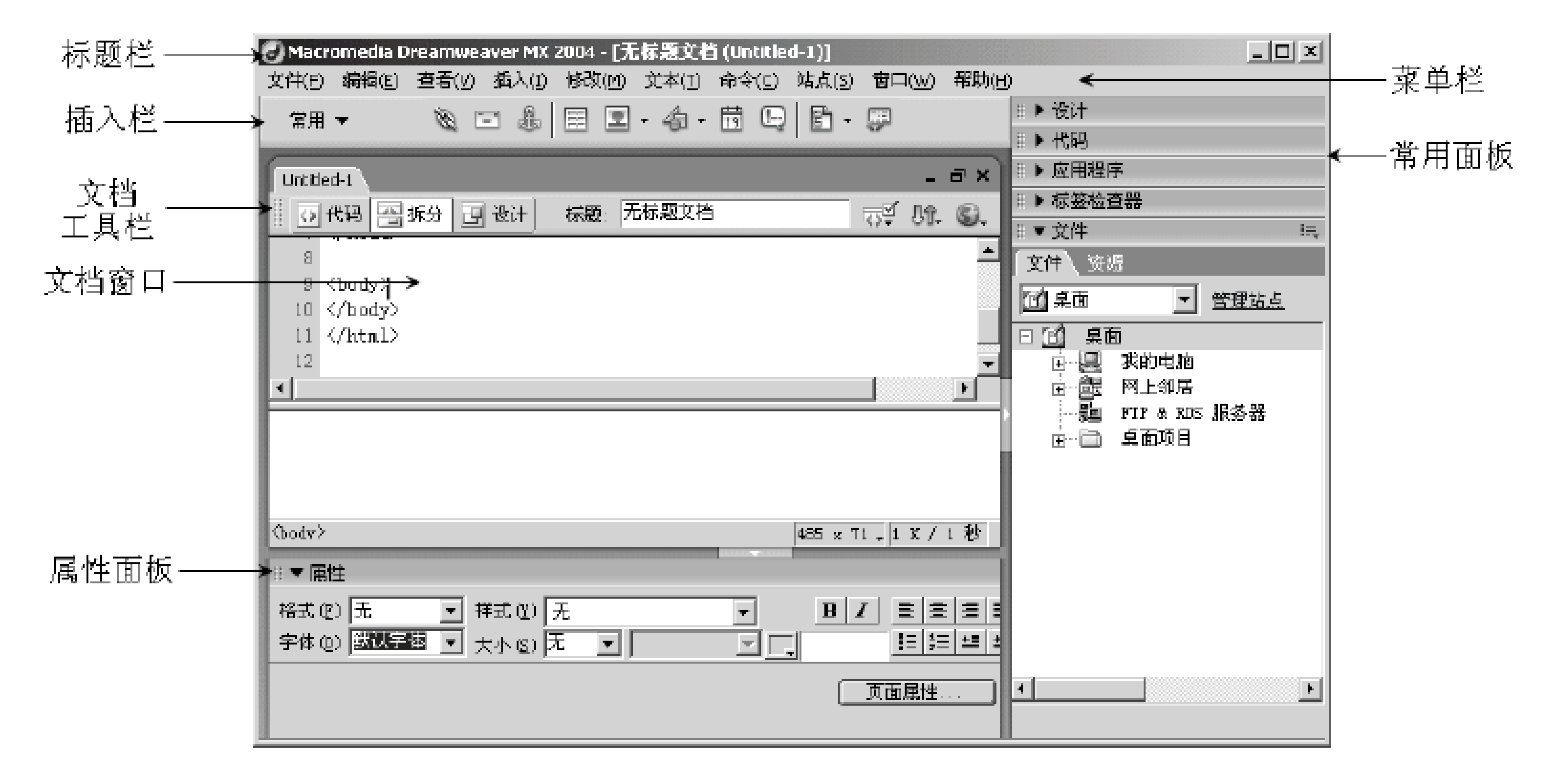


图 1-1　Dreamweaver MX 2004 的界面布局

1.4.1　标题栏

标题栏主要包括 Dreamweaver 标记、应用程序的名称、当前正在编辑文档的标题和名称等信息，还包括最小化按钮、最大化按钮以及关闭按钮。

单击 Dreamweaver 标记可以打开系统菜单。在 Dreamweaver 标记后面显示程序名称，之后的中括号“[ ]”内是当前打开的文档的标题，小括号“（ ）”内是该文件的名称。每次新建一个文档，Dreamweaver 都会自动将文档标题命名为“无标题文档”，文件名定义为 Untitled-x。其中，文档的标题和文档的文件名称是不同的概念。文件的标题通常在文档中的<title>和</title>标记中，是在用浏览器打开文档时显示在浏览器窗口的标题栏上的名称，而文件的名称则是文档存储在磁盘上的文件名。

1.4.2　菜单栏

菜单栏位于标题栏的下方，它包括文件、编辑、查看、插入、修改、文本、命令、站点、窗口和帮助 10 个菜单项，如图 1-2 所示。

文件(F)　编辑(E)　查看(V)　插入(I)　修改(M)　文本(T)　命令(C)　站点(S)　窗口(W)　帮助(H)

图 1-2　菜单栏

下面简单介绍一下 Dreamweaver 中的各个菜单项。

- “文件”菜单：包含文件操作的标准菜单项，例如新建、打开、保存等。“文件”菜单还包含各种其他命令，用于查看当前文档或对当前文档执行操作，例如“在浏览器中预览”和“打印代码”。
- “编辑”菜单：包含文件编辑的标准菜单项，例如剪切、复制和粘贴等。“编辑”菜单还包括选择和搜索命令，即“选择父标签”与“查找和替换”，并且提供对键盘快捷键编辑器、标签库编辑器和参数选择编辑器的访问。
- “查看”菜单：用于选择文档的不同视图（设计视图和代码视图），并且可以用于显示或隐藏不同类型的页面元素和Dreamweaver工具。
- “插入”菜单：提供插入面板的各项，用于将各种对象插入文档。
- “修改”菜单：用于更改选定页面元素或项的属性。使用此菜单，可以编辑标签属性、更改表格和表格元素，并且为库和模板执行不同的操作。
- “文本”菜单：用于设置文本的各种格式。
- “命令”菜单：提供对各种命令的访问，包括根据格式参数选择设置代码格式的命令、创建相册的命令，以及使用Macromedia Fireworks优化图像的命令。
- “站点”菜单：包含站点操作菜单项，这些菜单项可用于创建、打开和编辑站点，以及管理当前站点中的文件。
- “窗口”菜单：提供对Dreamweaver中所有面板、检查器和窗口的访问。
- “帮助”菜单：提供对Dreamweaver帮助文档的访问，包括用于使用Dreamweaver以及创建对Dreamweaver 的扩展的帮助系统，并且包括各种语言的参考材料。

1.4.3　插入栏

对于一些经常使用的操作，从菜单项中选择不是很方便。插入栏是 Dreamweaver 提供的一些菜单命令的快捷方法，其按钮一般都和菜单中的命令相对应，运用插入栏可以更方便快捷。插入栏集成了

多种网页元素，包括图像、文字等，如图 1-3 所示。



图 1-3　插入栏

单击插入栏左边的向下箭头▼，可以选择不同的工具组，包括布局、表单、文本等，如图 1-4 所示。如果选择“显示为制表符”命令，则以传统的方式显示插入栏，如图 1-5 所示。

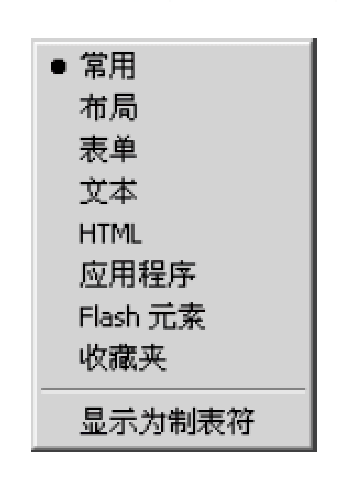


图 1-4　工具组菜单



图 1-5　传统方式的插入栏

“收藏夹”是用户自定义的组，在这里用户可以创建自己常用的按钮。

1.4.4　文档工具栏

“文档”工具栏包含按钮和弹出式菜单，它们提供各种“文档”窗口视图（如“设计”视图和“代码”视图）、各种查看选项和一些常用操作（如浏览器中预览），如图 1-6 所示。



图 1-6　文档工具栏

要显示文档工具栏，可以选择“查看”|“工具栏”|“文档”命令。文档工具栏中各个图标按钮的功能分别如下：

- （显示代码视图）：显示代码窗口。
- （显示代码和设计视图）：显示代码和设计窗口。
- （显示设计视图）：显示设计窗口。
- （活动数据视图）：用来查看实时数据视图。
- 标题: 无标题文档 （标题）：用来设置或修改文档的标题。
- （文件管理）：单击该按钮，通过这里来实现消除只读属性、获取、取出、上传、存回、撤销取出、设计注意以及站点定位等功能。
- （在浏览器中预览/调试）：单击该按钮在定义好的浏览器中预览或调试，或是编辑浏览器列表。
- （刷新）：单击该按钮，可以刷新设计显示中通过代码更改的部分。
- （参考）：单击该按钮来启动代码属性设置面板。
- （代码导航）：单击该按钮来移动文档里定义的JavaScript函数，设置或删除代码里的断点。

- （视图选项）（代码视图模式）：单击代码视图模式下的选项菜单按钮，可以选择自动换行、行数、高亮显示无效HTML、语法颜色、自动缩进等选项，以及在顶端查看设计视图。
- （视图选项）（设计视图模式）：单击设计视图模式下的选项菜单按钮，可以选择隐藏所有的可视化帮助或锁定可视化帮助，如表格边框、层边框、框架边框、图像地图、不可见元素、表头、标尺、网格以及轮廓图像。

1.4.5　属性面板

“属性”面板主要用于显示在文档窗口中所选中元素的属性，允许用户在属性面板中对元素属性直接进行修改。根据选中元素的不同，属性面板上的内容也不同。图 1-7、图 1-8 分别为文本和图像的属性面板。

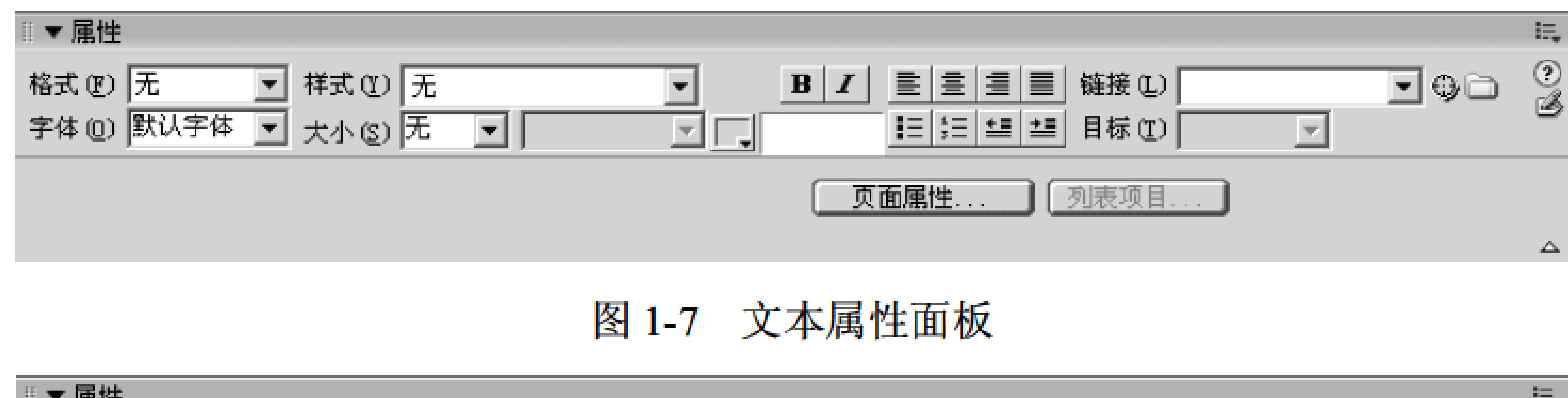


图 1-7　文本属性面板

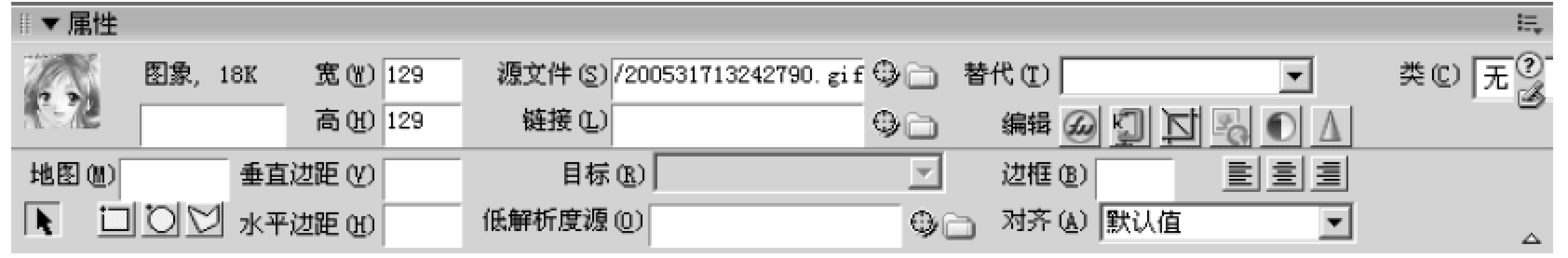


图 1-8　图像属性面板

在属性面板的右下角，通常都有一个指向上方的三角形图标。单击该三角形图标可以折叠属性面板。当属性面板被折叠时，单击指向下方的三角形图标又可以重新展开属性面板，显示更多的属性设置内容。

单击属性面板右上角的图标，则可以打开属性面板菜单。在菜单中可以选择属性面板的显示模式。

1.4.6　其他面板

在 Dreamweaver 中提供了一系列面板，在窗口的右侧以面板组的形式显示出来，如图 1-9 所示。它是一个工具集合的控制框架，可以将各种窗口、面板放置在其中，组合成为选项卡的形式，以节省屏幕空间。在必要时还可以将这些浮动窗口、面板拖离，成为独立的可停靠浮动面板，如图 1-10 所示。

如果希望将某个面板同其他面板组合成多个选项卡的形式，可以单击面板右上角的图标，从打开的菜单中选择“将 XX 组合在”子菜单中一个面板进行组合，如图 1-11 所示。



图 1-9　面板组



图 1-10　单独的浮动面板



图 1-11　以选项卡形式组合面板

1.5　在 Dreamweaver 中直接编写 HTML

在 Dreamweaver 中可以直接书写 HTML 的代码。下面通过一个 HTML 小实例进行说明。

1.5.1　编写 HTML 代码

（1）启动 Dreamweaver MX 2004 程序。如果是第一次启动该程序，系统会弹出“工作区设置”对话框，让用户选择工作界面的风格，如图 1-12 所示。如果以后改变了主意，可以在“首选参数”对话框中设置成不同的工作区风格。

（2）选中右侧的“代码编写者”单选按钮，单击“确定”按钮以代码视图的形式打开一个新的 HTML 文件，如图 1-13 所示。



图 1-12　“工作区设置”对话框

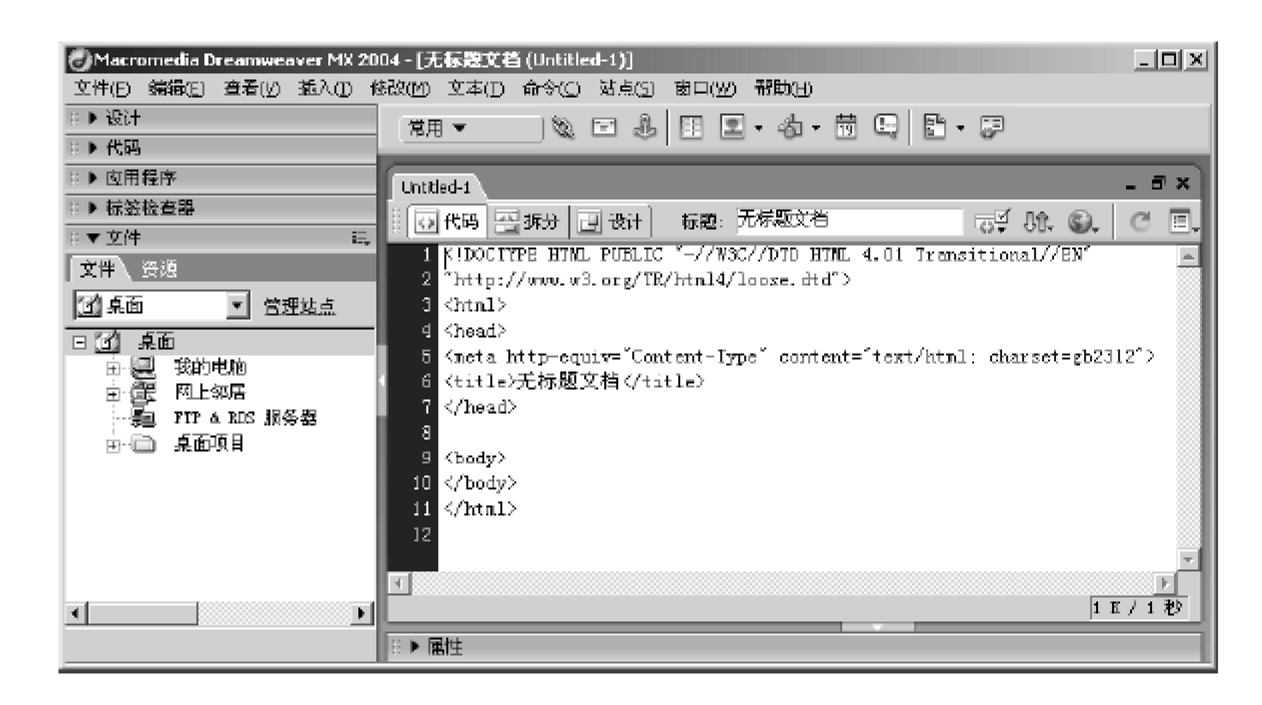


图 1-13　Dreamweaver 的代码视图

（3）在代码中<title>与</title>标签之间的内容就是新建 HTML 文件的标题，也就是将要在浏览器的标题栏中显示的页面标题。这里将其更改为“网页文件的标题”。

（4）在<body>与</body>标签之间添加主体内容的代码，如下：

```
<p>欢迎学习 HTML 语言！</p>
<p>我们愿做您最忠实的良师益友！</p>
<p>让我们伴您一起走入 HTML 的世界吧！！</p>
```

这样一个最基本的 HTML 文件就编写完成了。

1.5.2　保存并查看 HTML 文件

（1）选择“文件”|“保存”命令，打开如图 1-14 所示的“另存为”对话框。

（2）在“保存在”下拉列表框中选择存盘的位置，在“文件名”文本框中输入文件的名称“实例 1”，设置文件的保存类型为“HTML 文档”，单击 “保存”按钮完成文件的保存。

（3）双击保存的文件，可以在浏览器中看到文件的效果，如图 1-15 所示。



图 1-14　“另存为”对话框



图 1-15　实例 1 的页面效果

第 2 章

HTML 基本标记

- 头部标记——<head>
- 标题标记——<title>
- 元信息标记——<meta>
- 基底网址标记——<base>
- 页面的主体标记——body
- 页面注释标记——<!-- -->

本章介绍 HTML 的各种基本标记，这些都是一个完整的网页必不可少的。通过它们可以了解网页的基本结构及其工作原理。

2.1　头部标记——<head>

在 HTML 语言的头元素中，一般需要包括标题、基底信息、元信息等。HTML 的头元素是以<head>为开始标记，以</head>为结束标记的。一般情况下，CSS 和 JavaScript 都是定义在头元素中的，而定义在HTML语言头部的内容往往不会在网页上直接显示。它用于包含当前文档的相关信息，包括<title>、<base>、<basefont>、<isindex>、<meta>、<style>、<link>、<script>等。

下面介绍在 HTML 头部标记中的几种重要标记。

2.2　标题标记——<title>

HTML 页面的标题一般是用来说明页面用途的，它显示在浏览器的标题栏中。每个 HTML 页面都应该有标题，在 HTML 文档中，标题信息设置在页面的头部，也就是<head>与</head>之间。标题标记以<title>开始，以</title>结束。

语法：<title>…</title>

说明：在标记中间的“…”就是标题的内容，它可以帮助用户更好地识别页面。页面的标题有且只有一个，它位于 HTML 文档的头部，即<head>和</head>之间。

下面以实例说明标题标记的用法。

实例代码：

```
<html>
<head>
<title>文件的标题</title>
</head>
<body>
</body>
</html>
```

保存页面后在 IE 中打开，可以看到浏览器的标题栏中显示了刚才设置的标题“文件的标题”，效果如图 2-1 所示。



图 2-1　HTML 页面的标题

2.3　元信息标记——<meta>

meta 元素提供的信息是用户不可见的，它不显示在页面中，一般用来定义页面信息的名称、关键字、作者等。在 HTML 中，meta 标记不需要设置结束标记，在一个尖括号内就是一个 meta 内容，而在一个 HTML 头页面中可以有多个 meta 元素。meta 元素的属性有两种：name 和 http-equiv，其中 name 属性主要用于描述网页，以便于搜索引擎机器人查找、分类（目前几乎所有的搜索引擎都使用网上机器人自动查找 meta 值来给网页分类）。这其中最重要的是 description（站点在搜索引擎上的描述）和 keywords（关键词）。

下面根据功能的不同分别介绍元信息标记的使用方法。

2.3.1　设置页面关键字

设置页面关键字是为了向搜索引擎说明这一网页的关键词，从而帮助搜索引擎对该网页进行查找和分类，它可以提高被搜索到的几率，一般可设置不只 1 个关键字，之间用逗号隔开。但是由于很多搜索引擎在检索时会限制关键字数量，因此在设置关键字时不要过多，注意每一个关键字都要切中要害。

语法：<meta neme="keyword" content="具体的关键字">

说明：在该语法中，name 为属性名称，这里是“keyword”，也就是设置网页的关键字属性，而在 content 中则定义了具体的关键字的内容。

下面以实例来具体说明。

实例代码：

```
<html>
<head>
<title>学习元信息标记</title>
<meta name="keyword" content="html,元信息,关键字">
</head>
<body></body>
</html>
```

在实例中设定了“html”、“元信息”和“关键字”这 3 个词语作为该页面的关键字。

2.3.2　设置页面描述

设置页面描述也是为了便于搜索引擎的查找，它用来描述网页的主题等，与关键字一样，设置的页面描述也不会在网页中显示出来。

语法：<meta name="description" content="对页面的描述语言">

说明：在该语法中，name 为属性名称，这里设置为“description”，也就是将元信息属性设置为页面描述，在 content 中定义具体的描述语言。

在上一实例的基础上添加页面的描述信息，实例代码如下：

```
<html>
<head>
<title>学习元信息标记</title>
<meta name="keyword" content="html,元信息,关键字">
<meta name="description" content="关于 HTML 使用的网站">
</head>
<body></body>
</html>
```

在该实例中，设置了“关于 HTML 使用的网站”为网页的描述。

2.3.3　设置编辑工具

现在有很多编辑软件都可以制作网页，在源代码的头页面中可以设置网页编辑工具的名称。与其他 meta 元素相同，编辑工具也只是在页面的源代码中可以看到，而不会显示在浏览器中。

语法：<meta name="generator" content="编辑软件的名称">

说明：在该语法中，name 为属性名，设置为“generator”，也就是设置编辑工具，在 content 中定义具体的编辑器名称。

在上面实例的基础上添加编辑工具的信息，实例代码如下：

```
<html>
<head>
<title>学习元信息标记</title>
<meta name="keyword" content="html,元信息,关键字">
<meta name="description" content="关于 HTML 使用的网站">
<meta name="generator" content="Macromedia Dreamweaver MX 2004">
</head>
<body></body>
</html>
```

在这一实例中，以 Macromedia Dreamweaver MX 2004 作为网页的编辑工具。

2.3.4　设定作者信息

在源代码中还可以设置网页制作者的姓名信息。

语法：<meta name="author" content="作者的姓名">

说明：在该语法中，name 为属性名，设置为“author”，也就是设置作者信息，在 content 中定义具体的信息。

在上面实例的基础上添加作者的信息，实例代码如下：

```
<html>
<head>
<title>学习元信息标记</title>
<meta name="keyword" content="html,元信息,关键字">
```

```
<meta name="description" content="关于 HTML 使用的网站">
<meta name="generator" content="Macromedia Dreamweaver MX 2004">
<meta name="author" content="张三">
</head>
<body></body>
</html>
```

在这一实例中，将作者的姓名“张三”添加到了网页的源代码中。

2.3.5　限制搜索方式

网页可以通过在 meta 中的设置来限制搜索引擎对页面的搜索方式。

语法：<meta name="robots" content= "搜索方式">

说明：在该语法中，搜索方式的值和其所对应的含义见表 2-1。

表2-1　content值与对应的含义

content的值	含　义
all	页面将被检索，且页面上的链接可以被查询
none	页面不能被检索，且页面上的链接不可以被查询
index	页面将被检索，但页面上的链接却不可以被查询
follow	页面上的链接可以被查询
noindex	页面不能被检索，但页面上的链接可以被查询
nofollow	页面能被检索，但页面上的链接却不可以被查询

下面的实例中设定了网页能被检索，但页面上的链接却不可以被查询。

实例代码：

```
<html>
<head>
<title>限制搜索方式</title>
<meta name="robots" content="index ">
</head>
<body></body>
</html>
```

2.3.6　设置网页文字及语言

在网页中还可以通过语句来设定语言的编码方式。这样，浏览器就可以正确地选择语言，而不需要人工选取。

语法：<meta http-equiv="content-type" content="text/html; charset=字符集类型">
　　　<meta http-equiv="Content-Language" content="语言">

说明：在该语法中，http-equiv 用于传送 HTTP 通信协议的标头，也就是设定标头属性的名称，而

在 content 中才是具体的属性值。其中 charset 设置了网页的内码语系，也就是字符集的类型，中国内地常用的是 GB 码，charset 往往设置为 gb_2312，即简体中文。英文是 ISO-8859-1 字符集，此外还有 BIG5、utf-8、shift-Jis、Euc、Koi8-2 等字符集。如果采用第二种方法，则简体中文的设置为：

```
<meta http-equiv="Content-Language" content="zh_CN">
```

下面的实例中设置了语言为韩文，代码如下：

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=euc-kr">
<title>设定其他语言</title>
</head>
<body>
</body>
</html>
```

在运行该程序时，由于浏览器默认情况下并没有安装韩文，所以会出现需要安装语言的窗口，如图 2-2 所示。

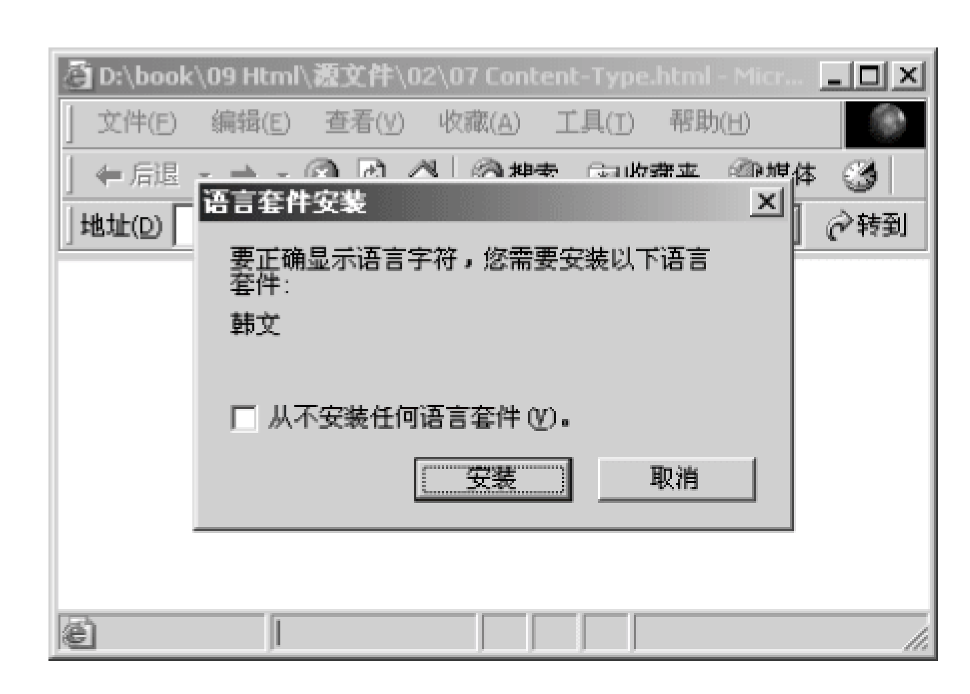


图 2-2　安装语言

2.3.7　设置网页的定时跳转

在浏览网页时经常会看到一些欢迎信息的界面，在经过一段时间后，这一页面会自动转到其他页面中，这就是网页的跳转。使用 HTTP 代码就可以很轻松地实现这一功能。

语法：<meta http-equiv="refresh" content="跳转时间;url=链接地址">

说明：在该语法中，refresh 表示网页的刷新，而在 content 中设定刷新的时间和刷新后的地址，时间和链接地址之间用分号相隔。默认情况下，跳转时间是以秒为单位的。

当链接地址为一个新的网页地址时，就会在设定的时间跳转到这个新的网址，其代码如下：

```
<html>
<head>
<title>学习元信息标记</title>
<meta http-equiv="refresh" content="3;url=http://www.sohu.com">
</head>
<body>
```

```
您好，本页在 3 秒之后将自动跳转到搜狐网站
</body>
</html>
```

运行程序，效果如图 2-3 所示。

在 3 秒之后，网页自动跳转到了搜狐网站，如图 2-4 所示。

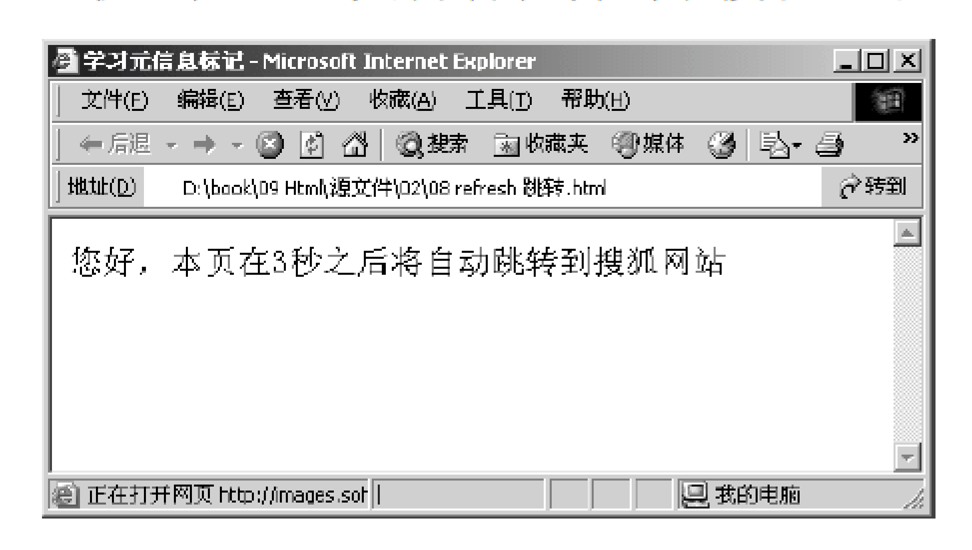




图 2-3　运行自动跳转的页面

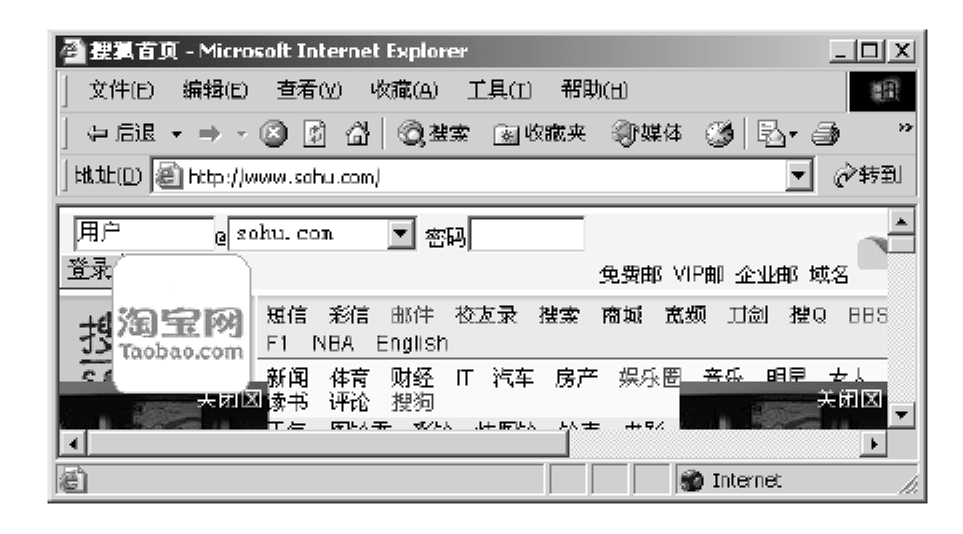


图 2-4　跳转后的页面

当语法中的链接地址被省略时，网页的功能就变成了刷新页面本身，这在不断更新数据的页面中常常会用到。刷新页面的代码如下：

```
<html>
<head>
<title>页面的刷新</title>
<meta http-equiv="refresh" content="60">
</head>
<body>
您好，本页每隔 1 分钟自动刷新一次
</body>
</html>
```

运行页面的效果如图 2-5 所示。

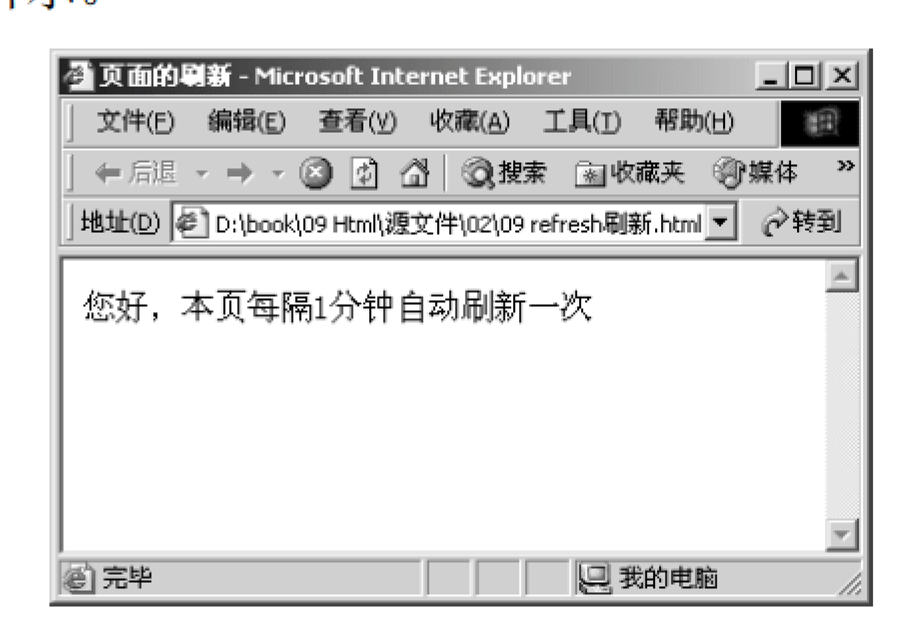


图 2-5　页面的刷新效果

2.3.8　设定网页的到期时间

在某些网站上会设置网页的到期时间，一旦过期则必须到服务器上重新调用。

语法：<meta http-equiv="expires" content="到期的时间">

说明：在该语法中，到期的时间必须是 GMT 时间格式。

实例代码：

```
<html>
<head>
<title>设置页面的到期时间</title>
<meta http-equiv="expires" content="Wed,27 July 2005 11:00:00 GMT">
</head>
<body>
</body>
</html>
```

在实例中设置了网页的到期时间为 2005 年 7 月 27 日 11 点整。

2.3.9　禁止从缓存中调用

浏览器一般为了节约时间，都在本地硬盘上保存一个网上的文件作为临时版本。在用户下次打开这个网页时，浏览器将会直接调用硬盘上的这个版本，而不是网上的。如果想让浏览者每次打开网页时都看到最新版本，那么就需要禁止浏览器调用缓存中的版本。如果在网页中设定了禁止缓存调阅，那么用户一旦离开网页，就无法从缓存中再调出该页面，而要到网络中查找该页面。

语法：<meta http-equiv="cache-control" content="no-cache">

<meta http-equiv="pragma" content="no-cache">

说明：在该语法中，cache-control 和 pragma 都可以用来设定缓存的属性，而在 content 中则是真正禁止调用缓存的语句。

实例代码：

```
<html>
<head>
<title>设置禁止调用缓存</title>
<meta http-equiv="cache-control" content="no-cache">
<meta http-equiv="pragma" content="no-cache">
</head>
<body>
</body>
</html>
```

2.3.10　删除过期的 cookie

如果网页过期，则删除存盘的 cookie。

语法：<meta http-equiv="set-cookie" content="到期的时间">

说明：在该语法中，到期的时间必须是 GMT 时间格式。

实例代码：

```
<html>
<head>
<title>删除过期的 cookie</title>
```

```
<meta http-equiv="set-cookie" content="Wed,27 July 2005 11:00:00 GMT">
<body>
</body>
</html>
```

在实例中，设置了网页的到期时间为 2005 年 7 月 27 日 11 点整，也就是这个时候删除存盘的 cookie。

2.3.11　强制打开新窗口

强制页面在当前窗口中以独立的页面显示，可以防止自己的网页被别人当作一个 frame 页调用。

语法：<meta http-equiv="windows-target" content="_top">

说明：在该语法中，windows-target 表示新网页的打开方式，而 content 中设置的_top 则代表打开的是一个独立页面。

实例代码：

```
<html>
<head>
<title>强制打开为独立页面</title>
<meta http-equiv="windows-target" content="_top">
<body>
</body>
</html>
```

2.3.12　设置网页的过渡效果

在浏览某些网站时，常常会在进入或退出某个网页时看到一些过渡效果，使得网页更加生动。这些效果在 meta 属性中就可以实现。

语法：<meta http-equiv="过渡事件" content="revealtrans(duration=过渡效果持续时间,transition=过渡方式)">

说明：在该语法中，过渡事件可以进入页面或者离开页面。进入页面的 http-equiv 值为 page-enter，离开页面的 http-equiv 值为 page-exist。过渡效果的持续时间默认情况下以秒为单位，过渡方式的编号见表 2-2。

表2-2　过渡方式的编号及含义

编　号	过 渡 方 式	编　号	过 渡 方 式
0	盒状收缩	12	随意溶解
1	盒状放射	13	从左右两端向中间展开
2	圆形收缩	14	从中间向左右两端展开
3	圆形放射	15	从上下两端向中间展开
4	由下往上	16	从中间向上下两端展开
5	由上往下	17	从右上角向左下角展开
6	从左至右	18	从右下角向左上角展开

续表

编　　号	过 渡 方 式	编　　号	过 渡 方 式
7	从右至左	19	从左上角向右下角展开
8	垂直百叶窗	20	从左下角向右上角展开
9	水平百叶窗	21	水平线状展开
10	水平格状百叶窗	22	垂直线状展开
11	垂直格状百叶窗	23	随机产生一种过渡方式

利用下面的实例来演示进入和离开网页的过渡效果，其中以随意溶解的效果进入页面 14enter.html，而以垂直线状展开离开页面 14_exit.html，过渡时间都是 3 秒。

（1）建立离开过渡效果的页面 14_exit.html，代码如下：

```
<html>
<head>
<title>页面的退出效果</title>
<meta http-equiv="page-exit" content="revealtrans(duration=3,transition=12)">
</head>
<body>
	<center>
		<img src="14/01.jpg" width="350" height="247"><br><br>
		<a href=14_page.html>退出页面</a>
	</center>
</body>
</html>
```

其显示效果如图 2-6 所示。

（2）建立中间过渡的页面 14_page.html，其代码如下：

```
<html>
<head>
<title>页面的过渡效果</title>
</head>
<body>
	<center>
		<Img src="14/02.jpg" width=350 height=247><br><br>
		<a href="14enter.html">进入网页</a>
	</center>
</body>
</html>
```

该页面的页面效果如图 2-7 所示。

（3）创建要显示进入过渡效果的页面 14enter.html，其代码如下：

```
<html>
<head>
<title>进入页面的过渡效果</title>
<meta http-equiv="page-enter" content="revealtrans(duration=3,transition=22)">
</head>
```

```
<body>
    <center>
        <img src="14/03.jpg" width=350 height=247><br><br>
        <a href=14_exit.html>返回</a>
    </center>
</body>
</html>
```

要显示进入过渡效果的页面 14enter.html 的页面效果如图 2-8 所示。



图 2-6　14_exit.html 的页面效果



图 2-7　过渡页的页面效果



图 2-8　14enter.html 的页面效果

（4）运行 14_exit.html，单击“退出页面”链接文字，可以看到如图 2-9 所示的退出页面的过渡效果。

（5）单击页面中的“进入网页”链接文字，可以打开页面 14enter.html，同时看到如图 2-10 所示的过渡效果。



图 2-9　退出页面的过渡效果

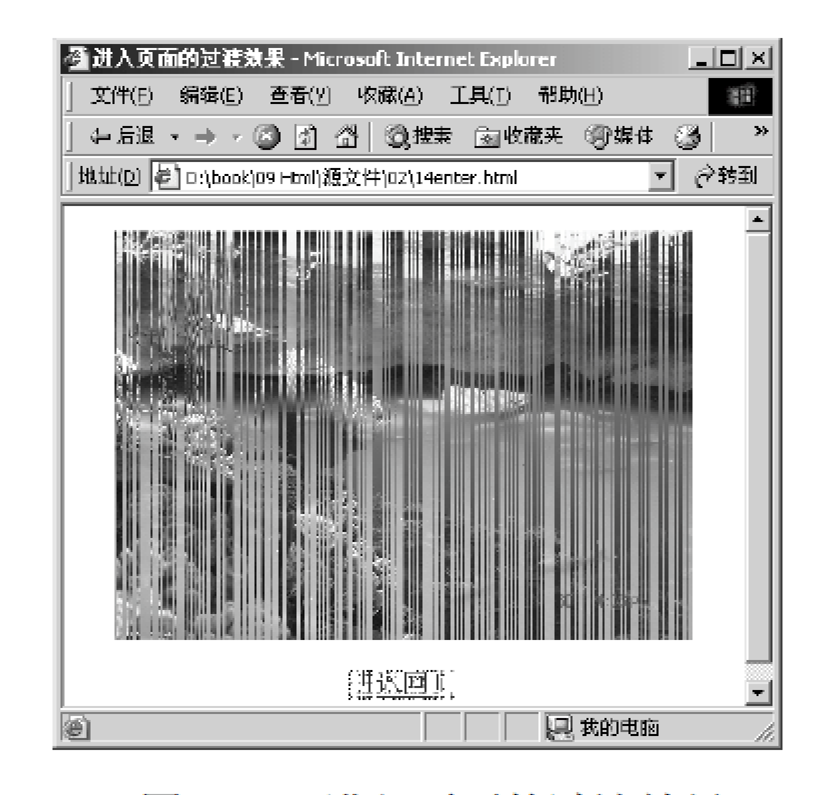


图 2-10　进入页面的过渡效果

2.4　基底网址标记——<base>

URL 路径是一种互联网地址的表示法，在这个数据里可以包括以何种协议连接、要连接到哪一个地址、连接地址的端口（Port）号以及服务器（Server）里页面的完整路径和页面名称等信息。在 HTML

中，URL 路径分为两种形式：绝对路经和相对路径。绝对路径是将服务器上磁盘驱动器名称和完整的路径写出来，同时也会表现出磁盘上的目录结构；相对路径是相对于当前 HTML 文档所在目录或站点根目录的路径。

HTML 页面通过基底网址把当前 HTML 页面中所有的相对 URL 转换成绝对 URL。一般情况下，通过基底网址标记<base>设置 HTML 页面的绝对路径，那么在页面中的链接地址只需设置成相对地址即可，当浏览器浏览页面时，会通过<base>标记将相对地址附在基底网址的后面，从而转化成绝对地址。

例如，在 HTML 页面的头部定义基底网址如下：

<base href="http://www.new.com/sample">

在页面主体中设置的某一个相对地址，如下：

<a href="../01/sample01.html">

当使用浏览器浏览时，这个链接地址就变成如下的绝对地址：

http://www.new.com/sample/01/sample01.html

因此，在 HTML 页面中设置基底标记时不应该多于一个，而且要将其放置在头部以及任何包含 URL 地址的语句之前。

语法：<base href="链接地址" target="新窗口的打开方式">

说明：在该语法中，“链接地址”就是要设置的页面的基底地址，而“新窗口的打开方式”可以设置为不同的效果，其属性值及含义见表 2-3。

表2-3　链接窗口的打开方式

属　性　值	打 开 方 式
parent	在上一级窗口打开，一般常用在分帧的框架页中
blank	在新窗口打开
self	在同一窗口打开，可以省略
_top	在浏览器的整个窗口打开，忽略任何框架

下面以实例说明该标记的使用方法，实例代码如下：

```
<html>
<head>
<base href="http://www.taobao.com" target="_blank">
<title>学习元信息标记</title>
</head>
<body>
      <a href="../15.html">打开一个相对地址</a>
</body>
</html>
```

运行该程序，当鼠标移动到链接文字上面时，可以看到在 IE 的状态栏中显示出其完整的链接地址，它是由代码中设置的基底地址加上程序中的相对地址组成的，如图 2-11 所示。

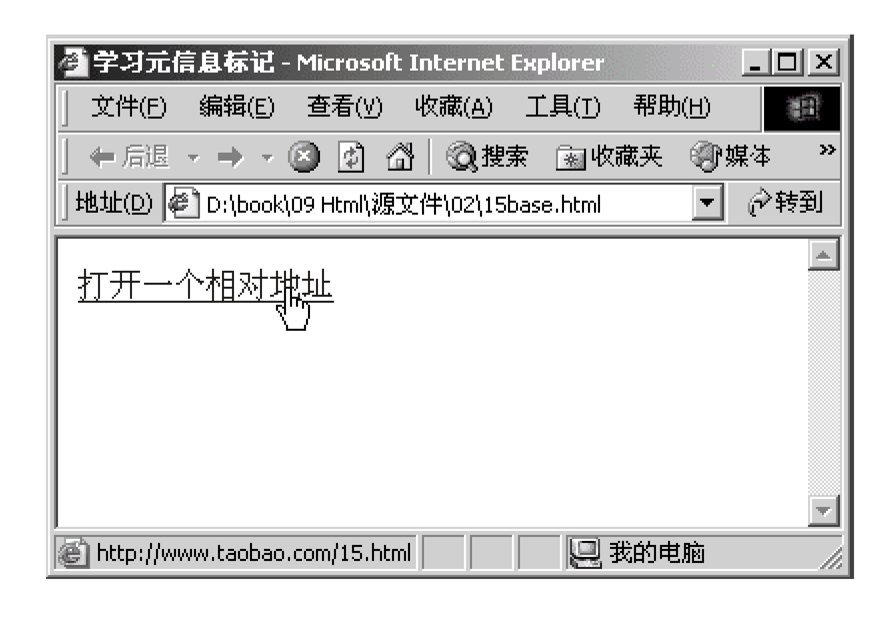


图 2-11　设置基底地址

2.5　页面的主体标记——body

网页的主体部分以<body>标记标志它的开始，以</body>标志它的结束。在网页的主体标记中有很多的属性设置，包括页面的背景设置、文字属性设置、链接设置、边距设置等。下面将逐步介绍这些网页主体标记的基本属性。

2.5.1　设置页面背景色——bgcolor

设置整个页面的背景颜色，需要用到 bgcolor 属性，它使用“#”加上 6 位的十六进制值来表现颜色。其中#FFFFFF 为白色，#000000 为黑色，#FF0000 为红色，#00FF00 为绿色，#0000FF 为蓝色。

语法：<body bgcolor="颜色代码">

说明：该语法中的 body 就是页面的主体标记，也就是说设置页面颜色要和页面的主体标记放置在一起。

下面的实例设置了页面的背景色为淡蓝色，其代码如下：

```
<html>
<head>
<title>设置页面背景色</title>
</head>
<body bgcolor="#3399CC">
</body>
</html>
```

运行这段代码，可以看到打开的页面背景为淡蓝色，颜色的值为#3399CC，效果如图 2-12 所示。



图 2-12　设置页面的背景颜色

2.5.2　设置背景图片——background

在网络上除了看到各种背景色的页面之外，还可以看到一些以图片作为背景的网页。使用恰当的图片作为背景能够使页面看上去更加生动美观。用图片作为背景需要使用 background 属性，还可以设置背景图片的平铺方式、显示方式等。

语法：<body background="文件链接地址" bgproperties="背景图片固定属性">

说明：文件的链接地址可以是相对地址，也就是本机上图片文件的存储位置，也可以设置为网上的图片资料，如 http://dvd.e0413.com/UPLOAD/IMGWSF/2005491443501.jpg。在默认情况下，用户可以省略 bgproperties 属性，这时图片会按照水平和垂直的方向不断重复出现，直到铺满整个页面。如果将 bgproperties 属性设置为 fixed，那么当滚动页面时，背景图像也会跟着移动，这相对浏览者来说，就是总停留在相同的位置上。

下面以实例说明背景图片的设置与显示效果。

（1）设置一个图片文件作为网页的背景，默认情况下不设置 bgproperties 属性，此时图片将在水

平和垂直方向平铺图像，代码如下：

```
<html>
<head>
<title>背景图片</title>
</head>
<body background="17/01.jpg">
</body>
</html>
```

运行这段代码，可以看到如图 2-13 所示的效果。图像在水平和垂直方向平铺。

（2）如果希望图片不重复显示，一般情况下需要借助 CSS 样式，这里简单介绍一下，在后面的章节中还将详细讲解 CSS 样式表的使用方法。

对于网页背景的样式设置，一般在头部标记中添加 style 标记，代码如下：

```
<html>
<head>
<title>背景图片</title>
<style type="text/css">
    body {background-repeat:no-repeat}
</style>
</head>
<body background="17/01.jpg">
</body>
</html>
```

在这段代码中，background-repeat 的值设置为 no-repeat，也就是不重复，运行效果如图 2-14 所示。如果在这段代码中，将 background-repeat 的值设置为 repeat-x，则背景图片只在水平方向平铺，效果如图 2-15 所示。相反，如果设置为 repeat-y，则只在垂直方向平铺。



图 2-13　平铺图像作为背景



图 2-14　背景图像单独显示



图 2-15　背景图像水平平铺效果

（3）除了设置背景是否重复之外，在网页中还可以设置背景图片是否变化。这一属性是通过 bgproperties 参数来设定的，将 bgproperties 的值设置为 fixed，背景图片会固定在页面上静止不动。其代码如下：

```
<html>
<head>
<title>背景图片</title>
```

```
</head>
<body background="17/02.jpg" bgproperties=fixed>
</body>
</html>
```

运行这段代码后的效果如图 2-16 所示。当拖动滚动条时，会发现只有文字在动，而背景是静止不动的，如图 2-17 所示。



图 2-16　运行代码的效果



图 2-17　拖动滚动条的效果

2.5.3　设置文字颜色——text

在页面中除了背景之外，对于默认文字的颜色设置可以通过 text 参数来实现。在没有对文字的颜色进行单独定义时，这一属性可以对页面中所有的文字起作用。

语法：<body text="颜色代码">

说明：在该语法中，text 的属性值与设置页面背景色相同，也就是说该属性设置也和在页面的主体标记放置在一起。

实例代码：

```
<html>
<head>
<title>设置页面文字颜色</title>
</head>
<body bgcolor="#99CCCC" text="#FF0000">
    设置页面的文字颜色
</body>
</html>
```

运行这段代码，实现的效果如图 2-18 所示。



图 2-18　设置页面文字颜色

2.5.4　设置链接文字属性——link

在网页创作中，除了文字、图片等，超链接也是最为常用的一种元素。超链接中以文字链接最多，在默认情况下，浏览器以蓝色作为超链接文字的颜色；访问过的文字则变为暗红色。用户在创作网页时，可以通过 link 参数修改链接文字的颜色。

语法：<body link="颜色代码">

说明：这一属性的设置与前面几个设置颜色的参数类似，都是与 body 标签放置在一起，表明它对网页中所有未单独设置的元素起作用。

（1）下面通过实例设置未访问的链接文字的颜色，代码如下：

```
<html>
<head>
<title>页面的链接文字</title>
</head>
<body text="#000000" link="#FF00FF">
    <center>
        设置文字的链接效果
        <br><br>
        <a href="http://www.yahoo.com">链接文字</a>
    <br><br>
</center>
</body>
</html>
```

运行这段代码，可以看到链接文字的颜色已经不是默认的蓝色，而是网页中设置的紫色，如图 2-19 所示。

（2）在这段代码的基础上，添加正在访问的文字颜色设置。这一属性需要用到 alink 参数，添加后的代码如下：

```
<html>
<head>
<title>页面的链接文字</title>
</head>
<body text="#000000" link="#FF00FF" alink="FF0000">
    <center>
        设置文字的链接效果
        <br><br>
        <a href="http://www.yahoo.com">链接文字</a>
        <br><br>
        <a href="http://www.taobao.com">正在访问的链接</a>
    </center>
</body>
</html>
```

运行这段代码之后，单击链接文字“正在访问的链接”，会发现按下鼠标时，文字颜色变成了红色，如图 2-20 所示。

（3）在这段代码的基础上修改一部分代码，使用 vlink 参数设置访问后的文字链接颜色，完成的代码如下：

```
<html>
<head>
<title>页面的链接文字</title>
</head>
```

```
<body text="#000000" link="#FF00FF" alink="FF0000" vlink="996600">
    <center>
        设置文字的链接效果
        <br><br>
        <a href="http://www.yahoo.com">链接文字</a>
        <br><br>
        <a href="http://www.huachu.com">已经访问过的链接</a>
    </center>
</body>
</html>
```

运行这段代码，会看到访问过的链接文字颜色变成了棕褐色，如图 2-21 所示。




图 2-19　设置链接文字的颜色



图 2-20　设置正在访问的文字颜色

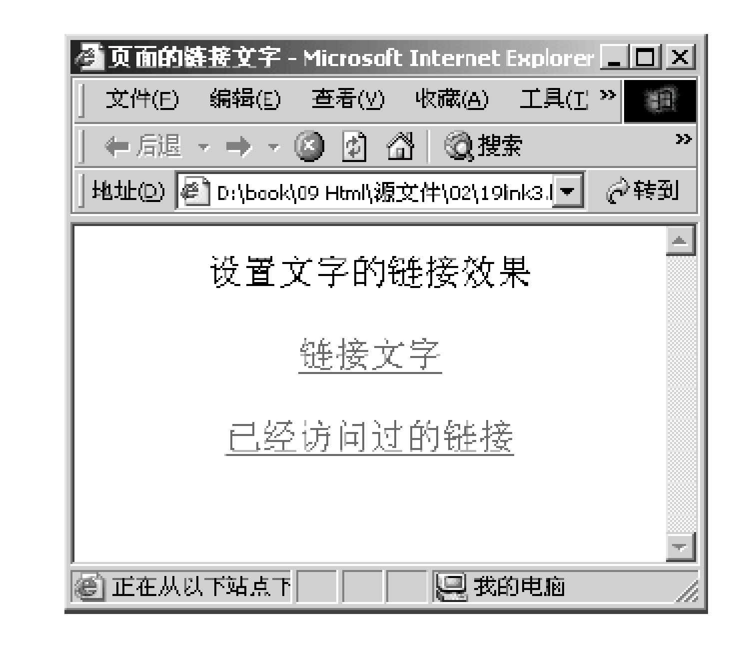


图 2-21　设置访问后的文字链接颜色

2.5.5　设置边距——margin

在网页的制作过程中，还可以定义页面的空白，也就是内容与浏览器边框之间的距离。其中包括上边框和左边框，其设置方法类似。

语法：<body topmargin=上边距的值 leftmagin=左边距的值>

说明：在默认情况下，边距的值是以像素为单位的，下面以实例说明设置边距的效果。

实例代码：

```
<html>
<head>
<title>设置边距</title>
</head>
<body topmargin=60 leftmargin=40>
    设置页面的上边距为 60 像素
    <br>
    设置页面的左边距为 40 像素
</body>
</html>
```

运行这段代码，可以看到设置边距前后的对比效果，设置边距前的效果如图 2-22 所示，设置自定义的边距效果如图 2-23 所示。

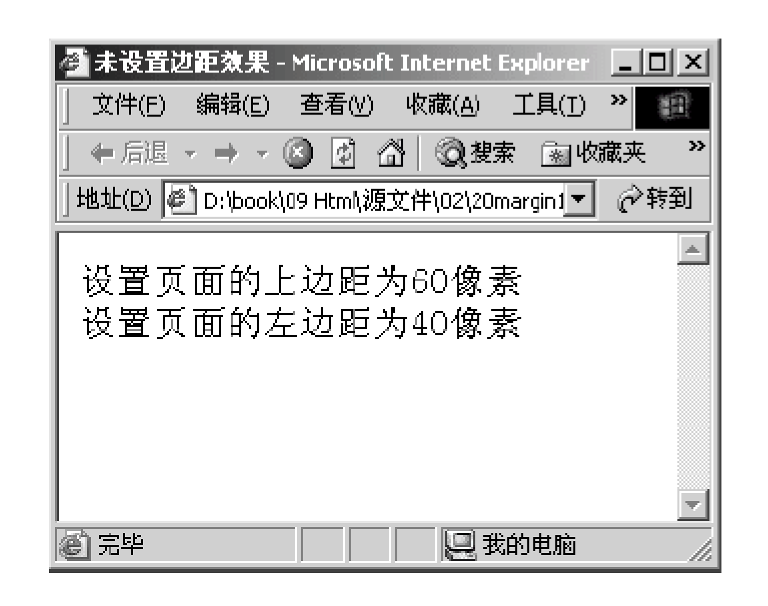


图 2-22　默认的页面效果

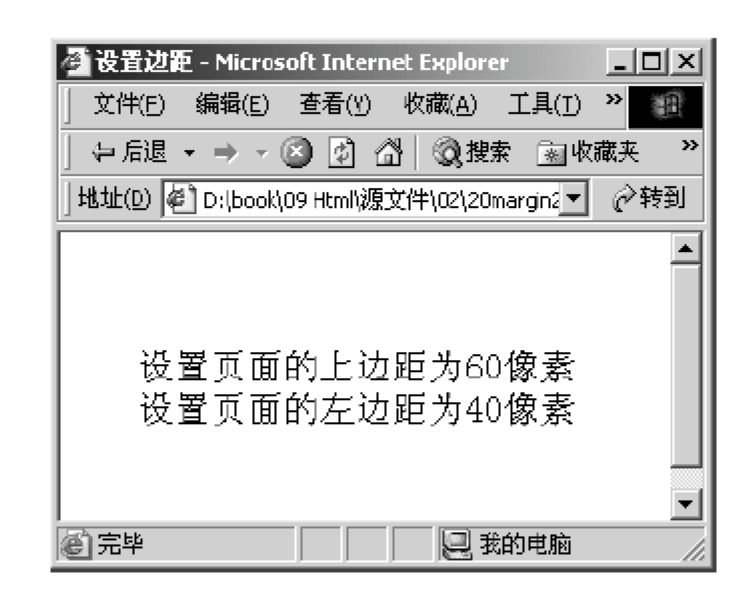


图 2-23　设置边距的效果

2.6　页面注释标记——<!--　-->

在网页中，除了以上这些基本元素外，还包含一种不显示在页面中的元素，那就是代码的注释文字。适当的注释可以帮助用户更好地了解网页中各个模块的划分，也有助于以后对代码的检查与维护，是一种很好的编程习惯。

语法：<!--注释的文字-->

说明：注释文字的标记很简单，只需要在语法中“注释的文字”的位置上添加需要的内容即可。

实例代码：

```
<html>
<head>
<title>设置代码的注释</title>
</head>
<body>
<!--居中显示-->
    <center>
        注释语句是用来帮助用户理解代码、维护代码的。<br>
        <!--超级链接-->
        <a href="http://www.microsoft.com">文字</a>
    </center>
</body>
</html>
```

在这段代码中，“居中显示”和“超级链接”这几个字就是对代码的注释，而代码所在行就是网页的注释语句，并不显示在浏览器中，效果如图 2-24 所示。

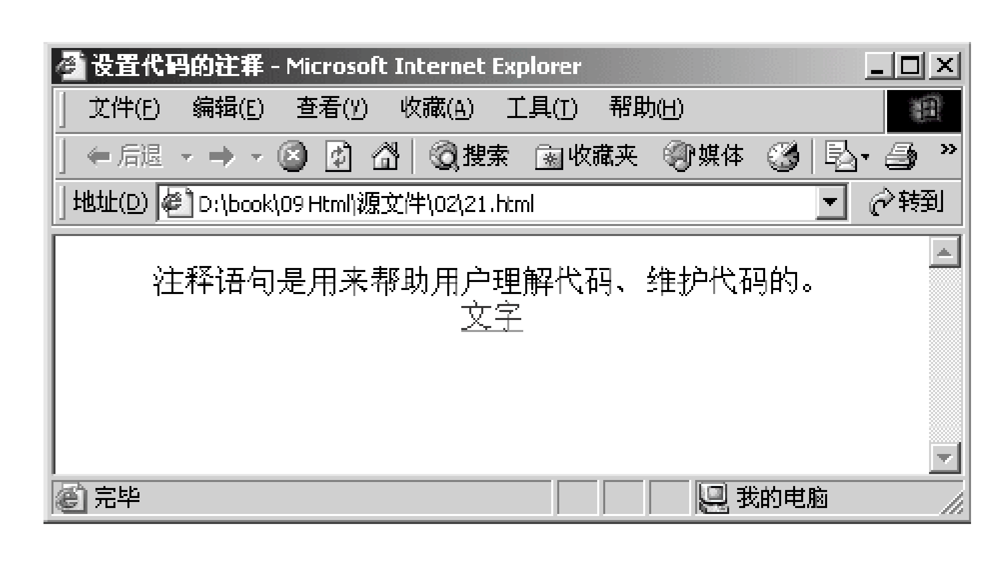


图 2-24　页面注释

第 3 章

文字与段落

- 标题文字的建立
- 文字格式标记
- 段落标记
- 水平线标记
- 其他标记

在网页创作中，文字是最基本的元素之一。增加文字的易读性，让浏览者在短时间内阅读更多、理解更多信息，并达到视觉艺术及传达的功能是网页创作者追求的目标。本章将介绍各种文字标记的使用方法。

3.1　标题文字的建立

在浏览网页时，常常看到一些标题文字，用于对文本中的章节进行划分，它们以固定的字号显示。HTML 文档中的标题文字分别用来指明页面上的 1～6 级标题。

3.1.1　标题文字标记

标题文字共包含 6 种标记，分别表示 6 个级别的标题，每一级别的字体大小都有明显的区别，从 1 级～6 级依次减小。

语法：

1 级标题：<h1>…</h1>

2 级标题：<h2>…</h2>

依次下去，到 6 级标题。

说明：在该语法中，1 级标题使用最大的字号表示；6 级标题使用的是最小的字号。

实例代码：

```
<!--这是关于标题文字的实例-->
<html>
<head>
<title>标题文字的效果</title>
</head>
<body>
    <h1>1 级标题的效果</h1>
    <h2>2 级标题的效果</h2>
    <h3>3 级标题的效果</h3>
    <h4>4 级标题的效果</h4>
    <h5>5 级标题的效果</h5>
    <h6>6 级标题的效果</h6>
</body>
</html>
```

运行这段代码可以看到网页中 6 种不同大小的标题文字，如图 3-1 所示。

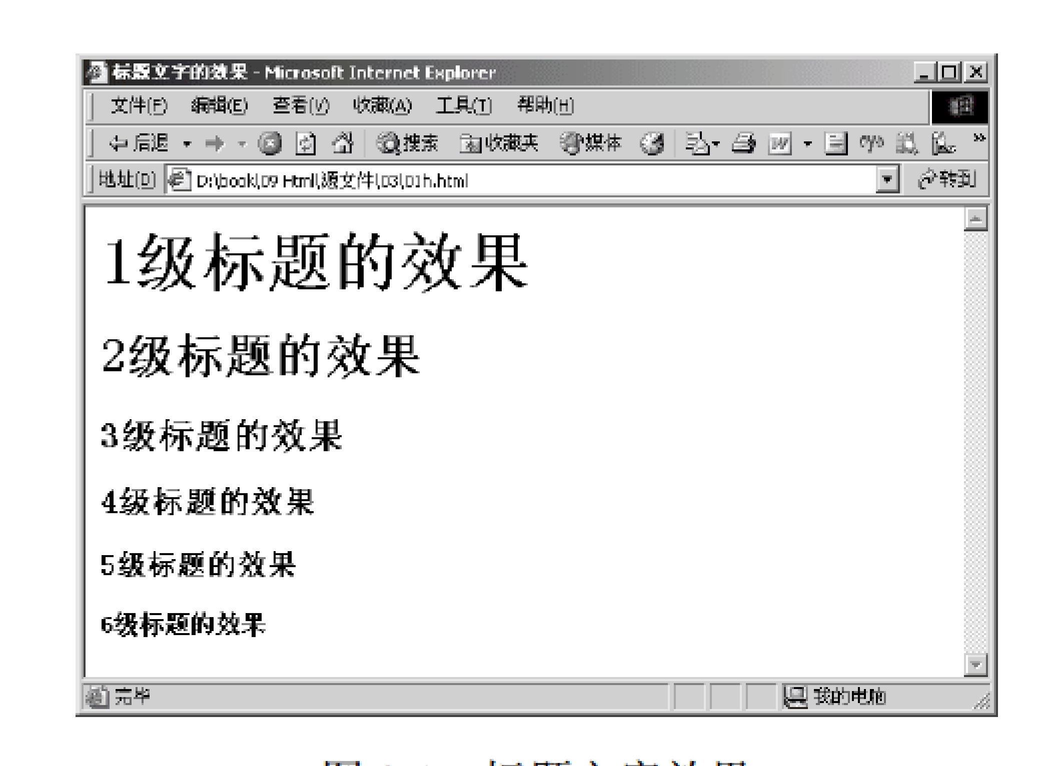


图 3-1　标题文字效果

3.1.2　标题文字的对齐方式——align

默认情况下，标题文字是左对齐的。而在网页制作的过程中，可以实现标题文字的编排设置。对于文字标题的属性设置中，最常用的就是关于对齐方式的设置，这需要使用 align 参数进行设置。

语法：align=对齐方式

说明：在该语法中，align 属性需要设置在标题标记的后面，其对齐方式的取值见表 3-1。

表3-1　标题文字的对齐方式

属　性　值	含　　义
left	左对齐
center	居中对齐
right	右对齐

实例代码：

```
<!--设置标题文字的不同对齐方式-->
<html>
<head>
<title>标题文字的对齐效果</title>
</head>
<body>
    <h1>1 级标题的默认对齐效果</h1>
    <h2 align=left>2 级标题的左对齐效果</h2>
    <h3 align=center>3 级标题的居中对齐效果</h3>
    <h4 align=right>4 级标题的右对齐效果</h4>
</body>
</html>
```

运行这段代码，可以看到不同对齐方式的标题效果，如图 3-2 所示。

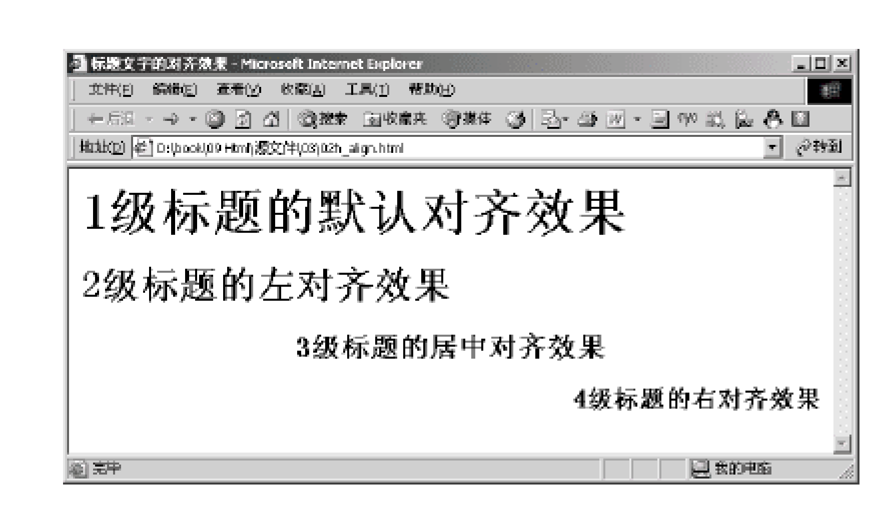


图 3-2　标题文字的对齐效果

3.2　文字格式标记

除了标题文字外，在网页中普通的文字信息更是不可缺少的。而多种多样的文字效果可以使网页变得更加绚丽。

在网页的编辑中，可以直接在文字的主体部分输入文字，而这些文字便会显示在页面中。可以说，这是 HTML 语言编辑中最简单的事情，只需要在<body>标记和</body>标记之间输入相应的文字即可。重要的是如何设置不同的文字效果，而这些属性的设置都位于文字格式标记中，下面将逐一进行讲解各种文字格式的设置方式。

3.2.1　设置文字字体——face

在 HTML 语言中，可以通过 face 属性设置文字的不同字体效果。而设置的字体效果必须在浏览器安装了相应的字体后才可以正确浏览，否则这些特殊字体会被浏览器中的普通字体所代替。因此，在网页中尽量减少使用过多的特殊字体，以免在用户浏览时无法看到正确的效果。由于浏览器默认情况下都包含了宋体、黑体等几种基本字体，因此网页创作者也应该注意在设计网页时，多利用这几种字体。

语法：应用字体的文字

说明：在该语法中，face 属性的值可以是 1 个或者多个。默认情况下，使用第 1 种字体进行显示；如果第 1 种字体不存在，则使用第 2 种字体进行代替，以此类推。如果设置的几种字体在浏览器中都不存在，则会以默认字体显示。

实例代码：

```
<!--设置不同的文字字体-->
<html>
<head>
<title>不同字体的显示效果</title>
</head>
<body>
    <font face="经典空叠圆简">经典空叠圆简的字体效果</font><br>
    <font face="黑体">黑体效果</font><br>
    <font face="Times New Roman,Times"> English fonts</font>
</body>
</html>
```

运行这段代码，可以看到几种不同的字体效果，如图 3-3 所示。



图 3-3　设置不同的文字字体

3.2.2　设置字号——size

除了字体外，文字的大小也是吸引用户注意的一个元素。除了使用标题文字标记设置固定大小的字号之外，HTML 语言提供了标记的 size 属性来设置普通文字的字号。

语法：

说明：在该语法中，文字的字号可以设置为 1~7，也可以是+1~+7 或者是-1~7。这些字号并没有一个固定的大小值，而是相对于默认文字大小来设定的，默认文字的大小与 3 号字相同，而数值越大，文字也越大。

实例代码：

```
<!--设置不同的文字大小-->
<html>
<head>
<title>不同字号文字的效果</title>
</head>
<body>
    <font size="1">1 号字体的效果</font><br>
    <font size="2">2 号字体的效果</font><br>
    <font size="3">3 号字体的效果</font><br>
    <font size="4">4 号字体的效果</font><br>
    <font size="5">5 号字体的效果</font><br>
    <font size="6">6 号字体的效果</font><br>
    <font size="7">7 号字体的效果</font><br>
    <font size="+2">默认字号+2，也就是 5 号字体的效果</font><br>
    <font size="-1">默认字号-1，即 2 号字体的效果</font>
</body>
</html>
```

运行这段代码，可以看到文字的大小变化，其效果如图 3-4 所示。

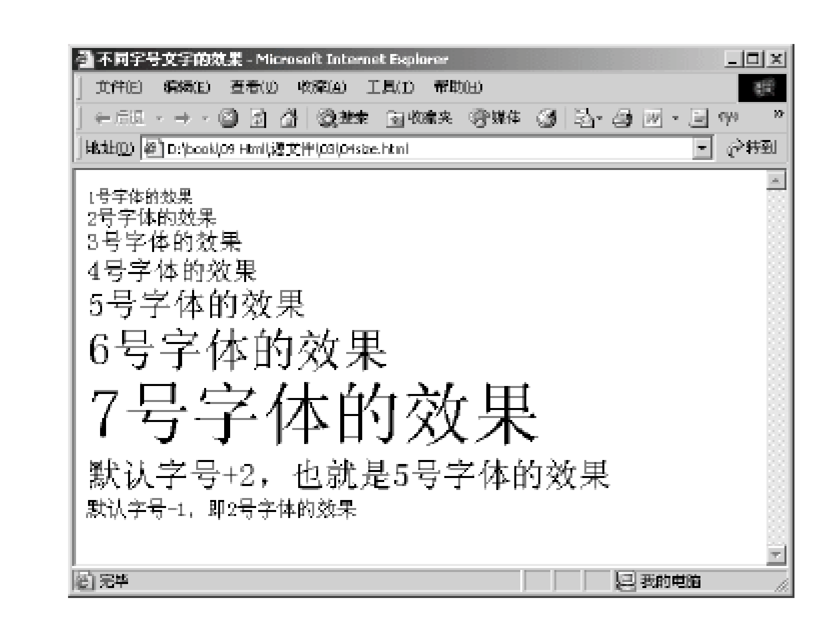


图 3-4　设置不同的字号

3.2.3　设置文字颜色——color

在 HTML 页面中，还可以通过不同的颜色表现不同的文字效果，从而增加网页的亮丽色彩，吸引浏览者的注意。

语法：

说明：与网页背景色的设置类似，颜色代码也是十六进制的。

实例代码：

```
<!--设置不同的文字颜色-->
<html>
<head>
<title>不同色彩的文字效果</title>
</head>
<body>
    <font face="黑体"color="#0099FF">淡蓝色的黑体效果</font><br>
    <font size="2" color="#990000">暗红色的 2 号文字效果</font><br>
    <font face="经典空叠圆简" size="5" color="#FF6600">桔色的 5 号
经典空叠圆简</font>
</body>
</html>
```

运行这段代码，可看到不同色彩的文字效果，如图 3-5 所示。

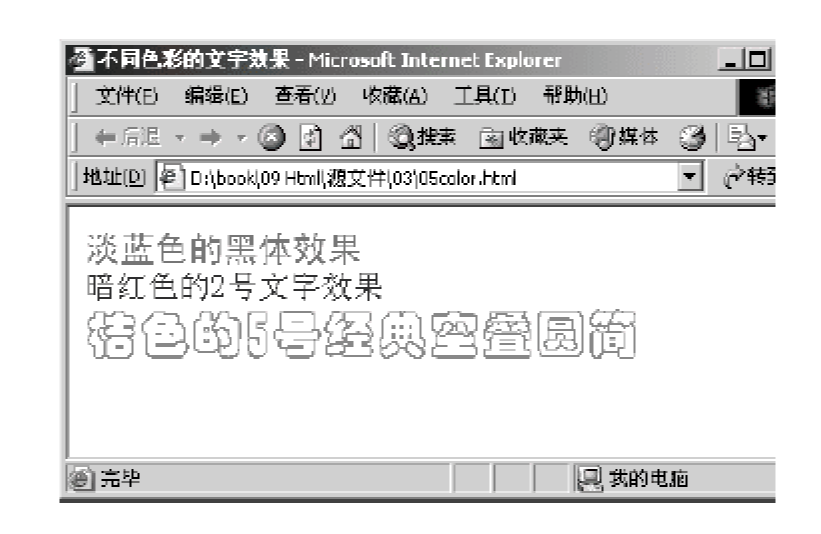


图 3-5　设置不同的文字颜色

3.2.4　粗体、斜体、下划线——strong、em、u

在浏览网页时，还常常可以看到一些特殊效果的文字，例如粗体字、斜体字以及下划线文字。而这些文字效果也可以通过设置 HTML 语言的标记来实现。

语法：粗体的文字

斜体字

<u>带下划线的文字</u>

说明：这几种效果的语法类似，只是标记不同。粗体的效果也可以通过标记<b>来实现；斜体字也可以使用标记<I>或者<cite>表示。

实例代码：

```
<!--设置不同的文字效果-->
<html>
<head>
<title>不同的文字样式</title>
</head>
<body>
    正常的文字效果<br><br>
    <strong>使用 strong 标记加粗文字</strong><br>
    <b>使用 B 标记加粗文字</b><br><br>
```

```
        <em>使用 em 标记的斜体效果</em><br>
        <I>使用 I 标记的斜体效果</I><br>
        <cite>使用 cite 标记的斜体效果</cite><br><br>
        <u>下划线文字效果</u><br>
</body>
</html>
```

运行这段代码，可以看到不同的样式效果，且使用不同的标记也可以达到相同的效果，如图 3-6 所示。

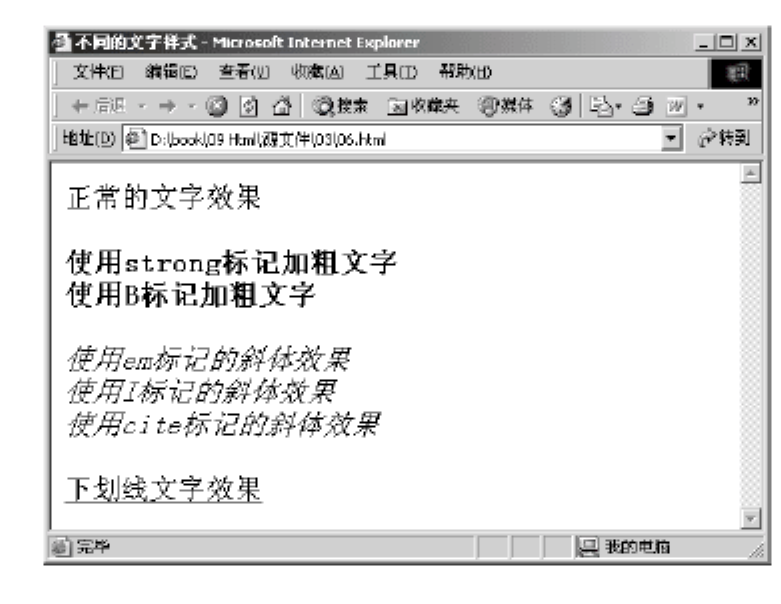


图 3-6　设置文字的不同样式

3.2.5　上标与下标——sup、sub

除了设置不同的文字效果之外，有时候在网页中还需要一种特殊的文字效果，即上标和下标，这在显示公式时常常会出现。在 HTML 语言中，也可以通过标记轻松地进行设置。

语法：[…]　　　　上标标记

　　　_…　　　　下标标记

说明：在该语法中，上标标记和下标标记的使用方法基本相同，只需要将文字放在标记中间即可。

实例代码：

```
<!--使用上标和下标-->
<html>
<head>
<title>上标与下标的效果</title>
</head>
<body>
        正常的文字效果<br><br>
        在方程式中应用上标的效果<br>
        X<sup>3</sup>+ 5X<sup>2</sup>-3=0<br><br>
        在文字中应用下标的效果<br>
        X<sub>1</sub>-Y<sub>2</sub>=11<br><br>
</body>
</html>
```

运行这段代码，可以看到在网页中显示了上标与下标的效果，如图 3-7 所示。

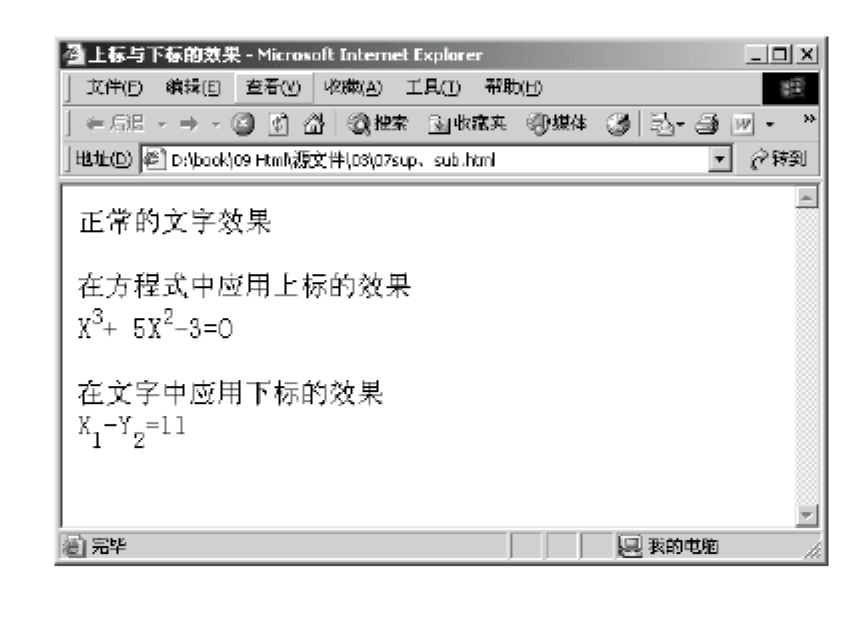


图 3-7　设置文字的上标与下标

3.2.6　设置删除线——strike

在网页中可以通过 strike 参数对文字添加删除线效果。

语法：<strike>文字</strike>　或<s>文字</s>

说明：这两种标记都可以创建删除线效果，使用起来也很简单，只要把需要设置成删除线效果的文字放置在标记中间即可。

实例代码：

```
<html>
```

```
<head>
<title>文字的删除线效果</title>
</head>
<body>
    正常的文字效果<br><br>
    在文字上使用 s 标记添加删除线<br>
    <s>文字的删除效果</s><br><br>
    在文字上使用 strike 标记添加删除线<br>
    <strike>文字的删除效果</strike>
</body>
</html>
```

运行这段代码，可以看到如图 3-8 所示的效果。

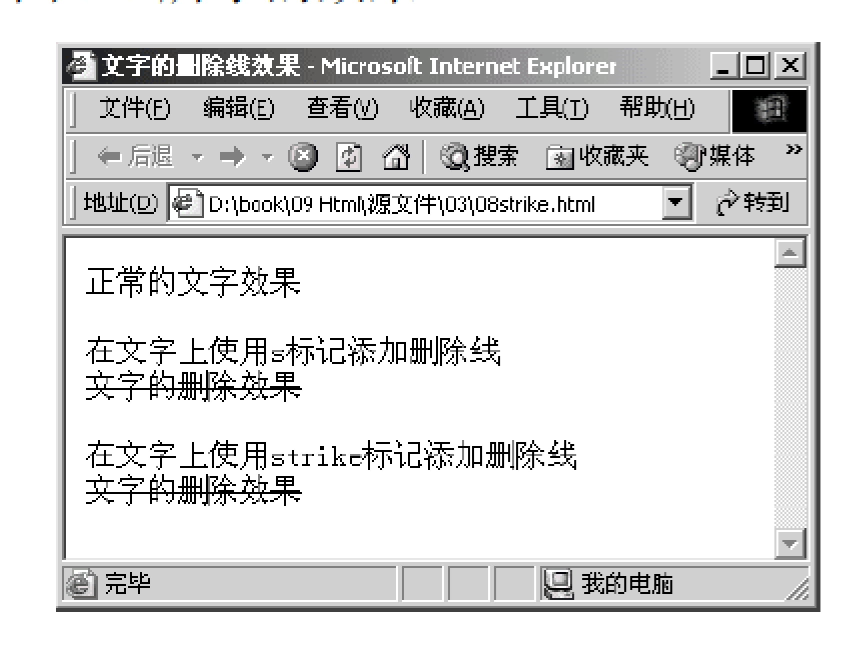


图 3-8　删除线效果

3.2.7　等宽文字标记——code

等宽文字标记常用于英文效果，使用该标记可以实现网页中字体的等宽效果。有时，使用等宽效果能够令页面显得更加整齐。

语法：<code>文字</code>

　　　<samp>文字</samp>

说明：在该语法中的这两种标记都可以实现文字的等宽显示，而在应用时只要把需要等宽显示的文字放置在标记中间即可。

实例代码：

```
<!--设置等宽文字-->
<html>
<head>
<title>文字的等宽效果</title>
</head>
<body>
    在下面将显示两段英文效果，突出等宽文字与普通英文文字的对比效果。<br><br>
    普通英文效果<br>
    <!--下面这段文字使用了正常的效果显示-->
    HTML is the lingua franca for publishing hypertext on the World Wide Web.<br><br>
    等宽文字效果<br>
```

```
        <!--下面这段英文使用了等宽文字的效果显示-->
        <code>HTML is the lingua franca for publishing hypertext on the World Wide Web.</code><br><br>
</body>
</html>
```

运行这段代码，可以看到如图 3-9 所示的效果。

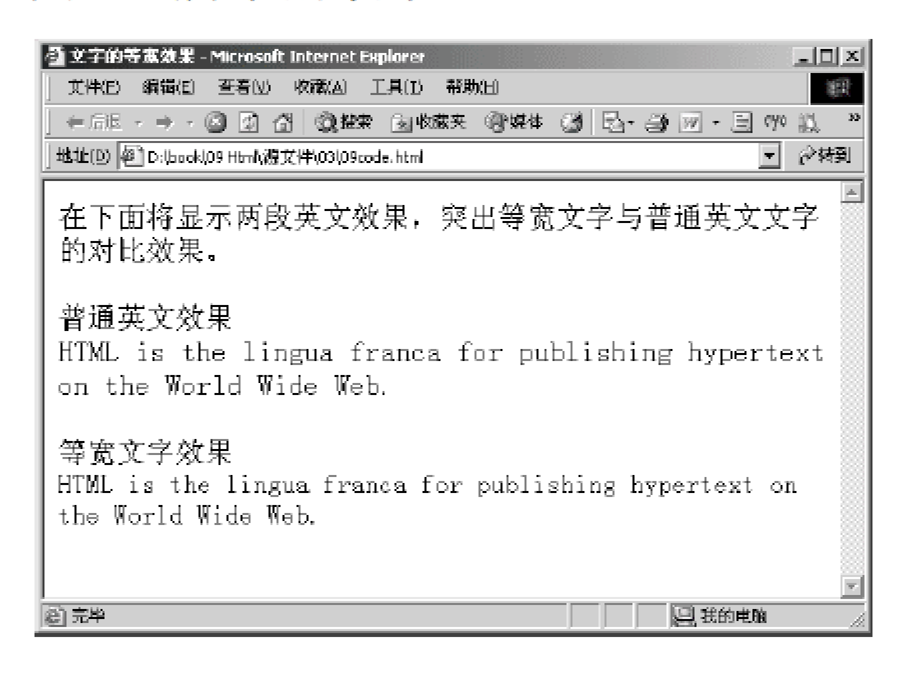


图 3-9　等宽文字的效果

3.2.8　空格——

一般情况下，在网页中输入文字时，如果在段落开始增加了空格，在使用浏览器进行浏览时往往看不到这些空格。这是因为在 HTML 文件中，浏览器本身会将两个句子之间的所有半角空白仅当作一个来看待。如果需要保留空格的效果，一般需要使用全角空格符号，或者通过空格码来代替。这里介绍如何应用空格码来输入保留文字中的空格效果。

语法：

说明：在网页中可以有多个空格，一个 只代表一个半角空格，多个空格则可以多次使用这一符号。

实例代码：

```
<html>
<head>
<title>输入空格符号</title>
</head>
<body>
        在段落开始输入空格符号的效果：<br>
            空格在网页排版中常常应用到，使用空格符号在文字的前方输入几个空格，就可
以实现首行缩进的效果。<br><br>
        在文字中间不使用空格符号，直接输入 6 个半角空格的效果：<br>
        白日依山尽，      黄河入海流。<br>
        欲穷千里目，      更上一层楼。<br><br>
        使用空格符号的效果：<br>
        白日依山尽，      黄河入海流。<br>
        欲穷千里目，      更上一层楼。<br>
</body>
</html>
```

运行这段代码，可以清楚地看到不管在两个句子间输入多少个半角空格，其中仅有一个半角的空格符会被接受，其余多出的空格符将被忽略掉。而输入空格代码则可以完整地保留空格的效果，如图 3-10 所示。

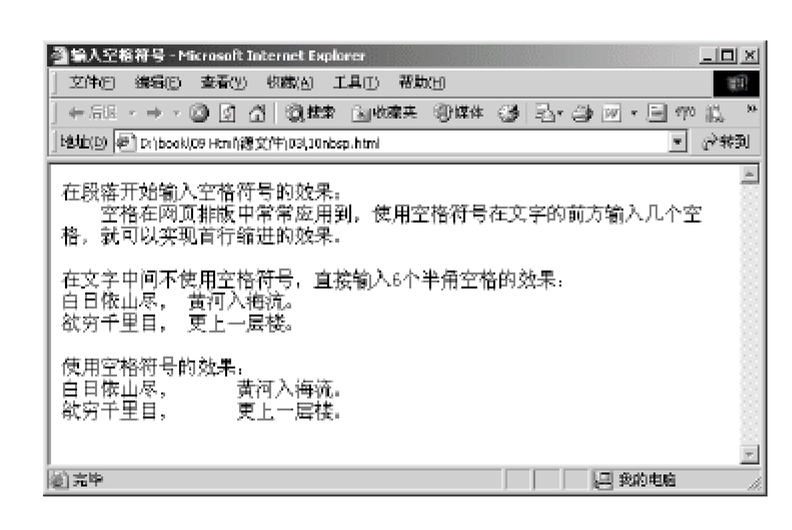


图 3-10　在网页中输入空格

3.2.9　其他特殊符号

除了空格以外，在网页的创作过程中，还有一些特殊的符号也需要使用代码进行代替。一般情况下，特殊符号的代码由前缀“&”、字符名称和后缀“;”组成。使用方法与空格符号类似，具体见表 3-2。

表3-2　特殊符号的表示

特 殊 符 号	符号的代码
“	"
&	&
<	<
>	>
×	×
§	§
©	©
®	®
™	™

说明：在需要输入这些特殊符号的位置时，使用相应的代码代替即可。

实例代码：

```
<html>
<head>
<title>输入特殊符号</title>
</head>
<body>
    引号：&quot;<br>
    左尖括号：&lt;<br>
    右尖括号：&gt;<br>
    乘号：&times;<br>
    小节符号：&sect;<br>
    版权所有的符号：&copy;<br>
```

```
        已注册的符号：&reg;<br>
        商标符号：&trade;<br>
</body>
</html>
```

运行这段代码，可以看到如图 3-11 所示的效果。

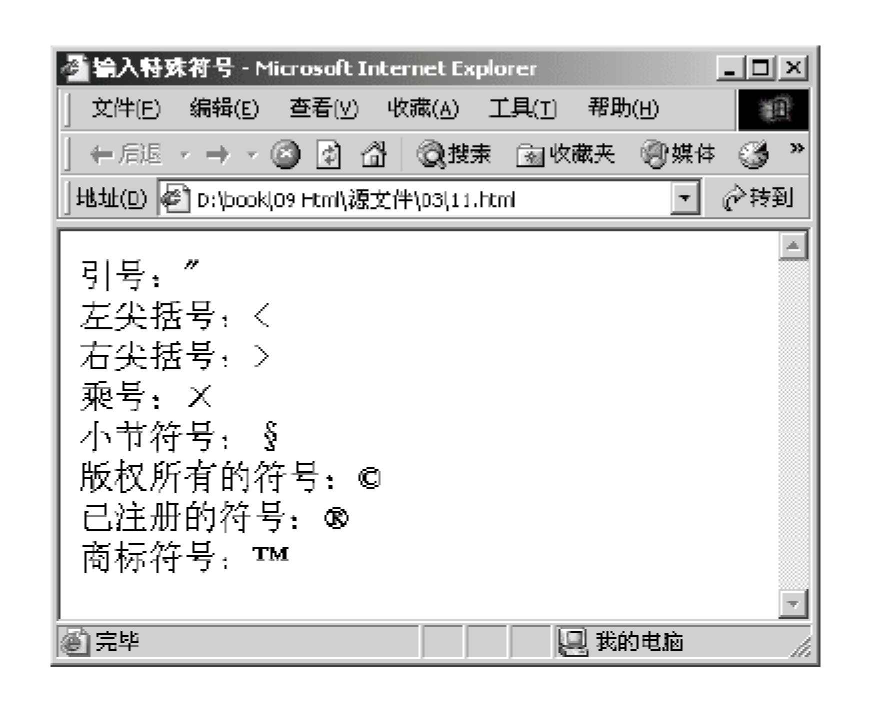


图 3-11　输入特殊符号

3.3　段落标记

在网页中如果要把文字有条理地显示，离不开段落标记的使用。在文本编辑窗口中，输入完一段文字后，按下回车键后就生成了一个段落。在 HTML 中可以通过标记实现段落的效果，下面具体介绍和段落相关的一些标记。

3.3.1　段落标记——p

在 HTML 语言中，段落通过<p>标记来表示。

语法：<p>段落文字</p>

说明：与其他标记不同的是，段落标记可以没有结束标记</p>，而每一个新的段落标记的开始同时也意味着上一个段落的结束。

实例代码：

```
<html>
<head>
<title>输入段落文字</title>
</head>
<body>
        <p>小镇上七千多人依水而居，镇上的主要街道有 9 条。
        <p>临河筑的民房、黑瓦白墙，屋脊起翘，鳞次栉比，古色古香，具有“人家尽忱河，水巷小桥多”的特色。</p>
        <p>又似一幅古雅、秀丽的水乡风情画。</p>
</body>
</html>
```

运行这段代码，可以看到两种方法的段落标记都可以成功地将文字分段。效果如图 3-12 所示。

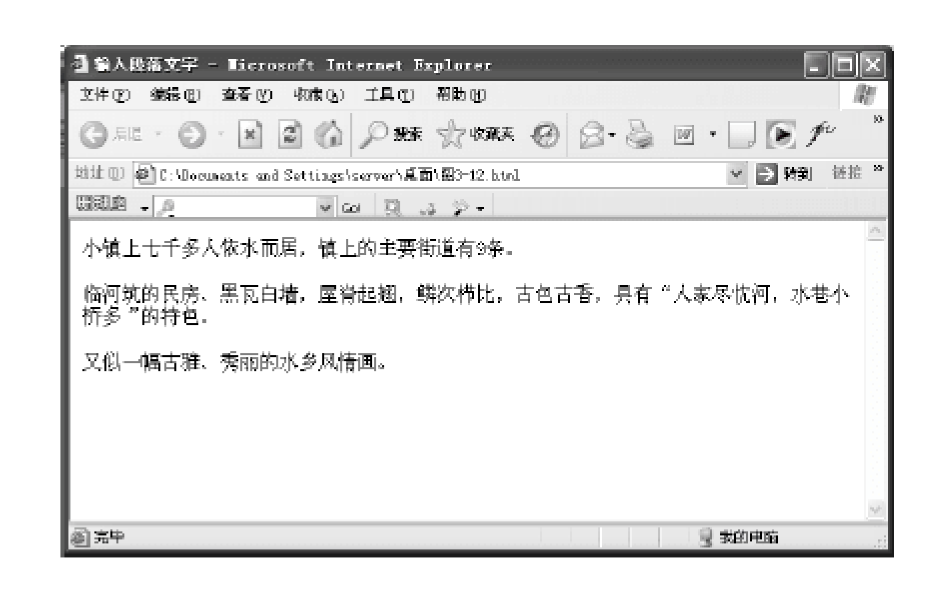


图 3-12　段落效果

3.3.2　取消文字换行标记——nobr

在网页的显示过程中，如果某一行的文字宽度过长，浏览器会自动对这段文字进行换行处理。如果用户在创作时不希望被自动换行，则可以通过 nobr 属性来实现。

语法：<nobr>不换行显示的文字</nobr>

说明：在标记之间的文字在显示的过程中不会自动换行。

实例代码：

```
<html>
<head>
<title>文字不换行显示</title>
</head>
<body>
    <!--当浏览器宽度不够时，文本内容会自动换行显示-->
    <p>HTML 的英文全称是 Hyper Text Markup Language，直译为超文本标记语言，它是全球广域网上描述网
页内容和外观的标准。</p>
    <!--下面这段文字不会自动换行显示，当浏览器宽度不够时，会出
现滚动条-->
    <p><nobr>HTML 的英文全称是 Hyper Text Markup Language，
直译为超文本标记语言，它是全球广域网上描述网页内容和外观的标
准。</nobr></p>
</body>
</html>
```

运行这段代码，可以看到强制文字不换行的效果，如图 3-13 所示。

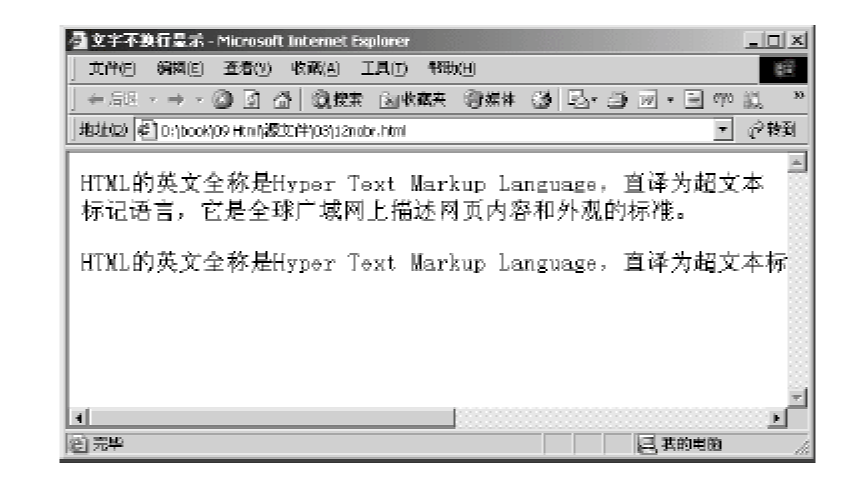


图 3-13　文字不换行的效果

3.3.3　换行标记——br

在网页的文字显示过程中，除了可以让文字不自动换行之外，还可以使用
标记将文字强制换行。这一换行标记与段落标记不同，段落标记的换行是隔行的，而使用换行标记能使两行的文字更加紧凑。

语法：

说明：一个
标记代表一个换行，连续的多个标记可以多次换行。

实例代码：

```
<html>
<head>
<title>文字的换行效果</title>
</head>
<body>
     下面是一段描写白龙池风景的文字：<br><br>
     在建岱桥北的溪谷内是著名的白龙池。<br>
     相传此处是东海龙王的小儿子小白龙在此潜居镇山治水。<br>
     这里上有百丈崖悬流下掷，似玉龙腾飞，顺着峡谷穿山越涧泻
入池内。
</body>
</html>
```

运行这段代码，可以看到使用换行标记的效果，如图 3-14 所示。

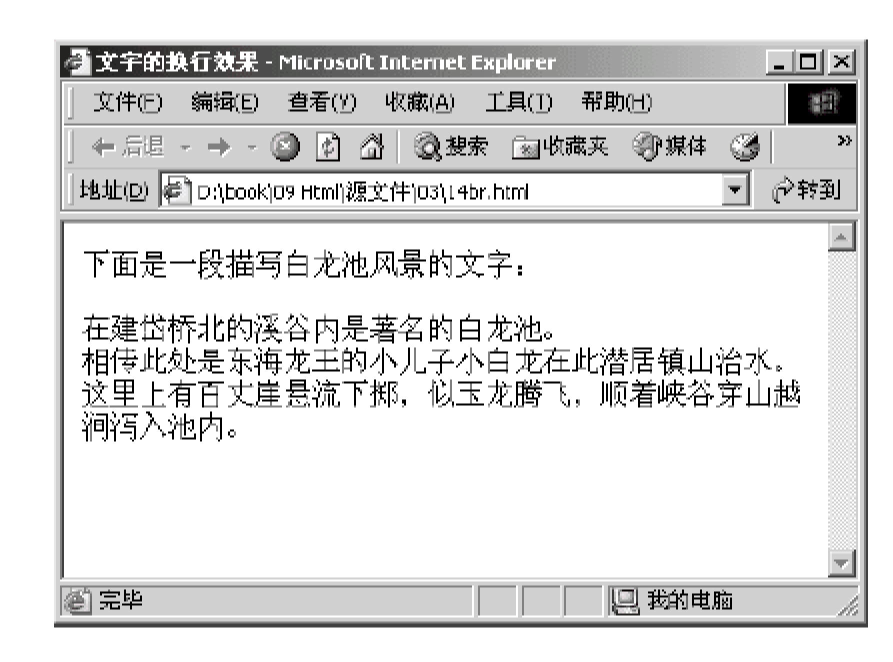


图 3-14　文字的换行

3.3.4　保留原始排版方式标记——pre

在网页创作中，一般是通过各种标记对文字进行排版的。但是在实际应用中，往往需要一些特殊的排版效果，这样使用标记控制起来会比较麻烦。解决的方法就是保留文本格式的排版效果，例如空格、制表符等。如果要保留原始的文本排版效果，则需要使用<pre>标记。

语法：<pre>文字</pre>

说明：在标记之间的文字会保留文档中的空白，按照原始的文本格式进行显示。

实例代码：

```
<!--使用标记保留文字的排版效果-->
<html>
<head>
<title>保留原始排版方式</title>
</head>
<body>
     <p>下面是原始文字的排版效果：</p>
     <pre>
            O  O                 K     K
         O        O              K   K
       O            O            K  K
      O              O           KK
       O            O            K  K
         O        O              K     K
            O  O                 K          K
     </pre>
</body>
</html>
```

运行这段代码，可以看到运行效果和文本中的效果相同，如图 3-15 所示。

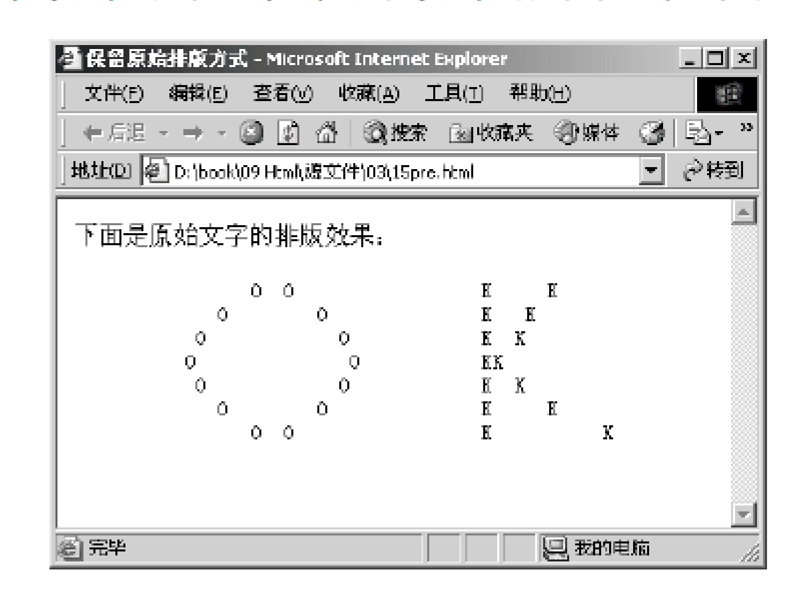


图 3-15　保留原始的排版效果

3.3.5　居中对齐标记——center

对于段落来说，和普通文字类似，有时候也需要将段落居中。在 HTML 语言中提供了专门的标记。

语法：<center>文字</center>

说明：在标记之间的文字会自动居中显示。

实例代码：

```
<html>
<head>
<title>居中对齐</title>
</head>
<body>
    <center>
        <p>春雪</p>
        新年都未有芳华，二月初惊见草芽。<br>
        白雪却嫌春色晚，故穿庭树作飞花。<br>
    </center>
</body>
</html>
```

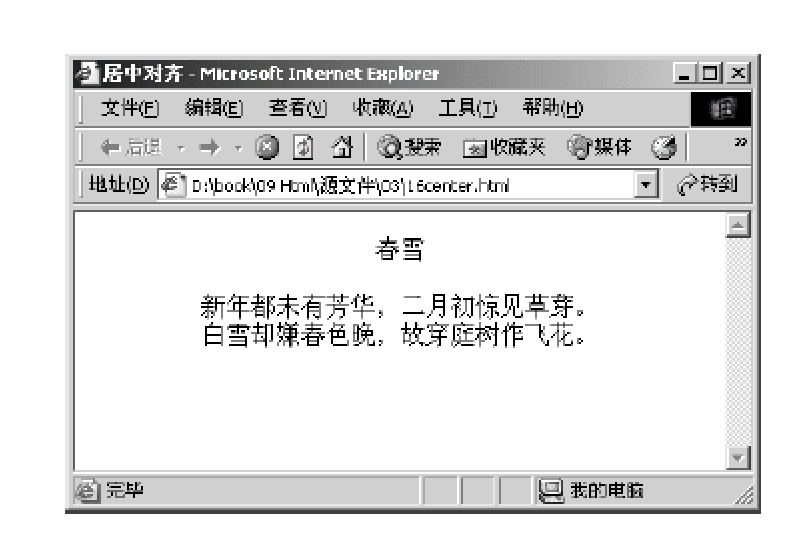


图 3-16　段落的居中显示

运行这段代码，可以看到这首古诗居中显示，如图 3-16 所示。

3.3.6　向右缩进标记——blockquote

使用<blockquote>标记可以实现页面文字的段落缩进。这一标记也是每使用一次，段落就缩进一次，可以嵌套使用，以达到不同的缩进效果。

语法：<blockquote>文字</blockquote>

说明：在该标记之间的文字会自动缩进。

实例代码：

```
<html>
<head>
<title>段落的缩进效果</title>
```

```
</head>
<body>
    相传，两千五百年前，春秋时期的大音乐家俞伯牙，曾学琴于程廉先生，三年不成。后来他沿着孔子的足迹
登游泰山，
    <blockquote>观东海日出，看云雾变化，</blockquote>
    <blockquote><blockquote>闻松风长啸，听水涛咆哮，</blockquote></blockquote>
    <blockquote><blockquote><blockquote>拜大自然为师，琴艺大有长进，
</blockquote></blockquote></blockquote>
    <blockquote><blockquote><blockquote><blockquote>写出了著名的古琴曲高山和流水。
</blockquote></blockquote></blockquote></blockquote>
</body>
</html>
```

运行这段代码，可以看到如图 3-17 所示的缩进效果。

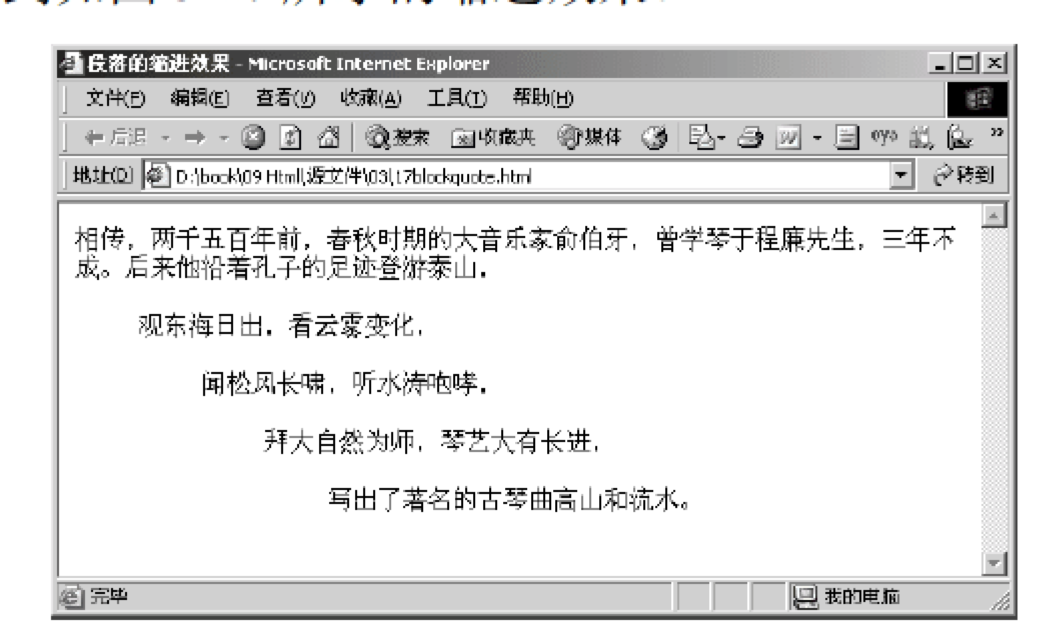


图 3-17　段落的缩进效果

3.4　水平线标记

在网页中常常看到一些水平线将段落与段落之间隔开，这些水平线可以通过插入图片实现，也可以更简单地通过标记来完成。

3.4.1　添加水平线——hr

语法：<hr>

说明：在网页中输入一个<hr>标记，就添加了一条默认样式的水平线。

实例代码：

```
<html>
<head>
<title>添加水平线</title>
</head>
<body>
    <center><h4>泰安：华夏文明发祥地之一</h4></center>
    <hr>
    <p>泰安是华夏文明发祥地之一。早在 50 万年前就有人类生存，5 万年前的新泰人已跨入智人阶段；5000
```

```
年前这里孕育了灿烂的大汶口文化，成为华夏文明史上的一个重要里程碑。</p>
</body>
</html>
```

运行代码，可以看到在网页中出现了一条水平线，如图 3-18 所示。

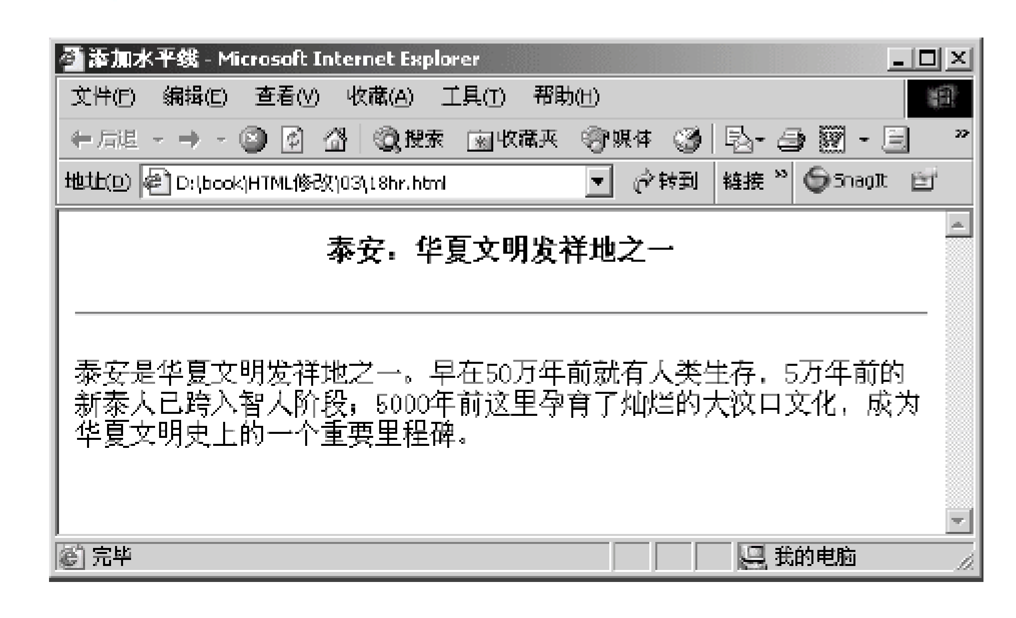


图 3-18　添加水平线

3.4.2　设置水平线宽度与高度属性——width、height

在默认情况下，在网页中插入的水平线是 100%的宽度，1 像素的高度。而在实际创建网页时，可以对水平线的宽度和高度进行设置。

语法：<hr width=水平线宽度 size=水平线高度>

说明：在该语法中，水平线的宽度值可以是确定的像素值，也可以是窗口的百分比。而水平线的高度值则只能够是像素数。如果在创建水平线时只设置一个参数，那么另外一个参数则会取默认值。

实例代码：

```
<html>
<head>
<title>设置水平线大小</title>
</head>
<body>
    <center>
        <font face="黑体" size=5>醉花阴</font>
        <hr width=130>
        <font size=4>李清照</font>
    </center>
    <hr width=85% size=3>
    <p>    薄雾浓云愁永画，瑞脑消金兽。<br>
        佳节又重阳，玉枕纱厨，半夜凉初透。<br>
        东离把酒黄昏后，有暗香盈袖。<br>
        莫道不消魂，帘卷西风，人比黄花瘦。</p>
    <hr size=5>
</body>
</html>
```

运行这段代码，可以看到 3 条高度和宽度不等的水平线效果，如图 3-19 所示。

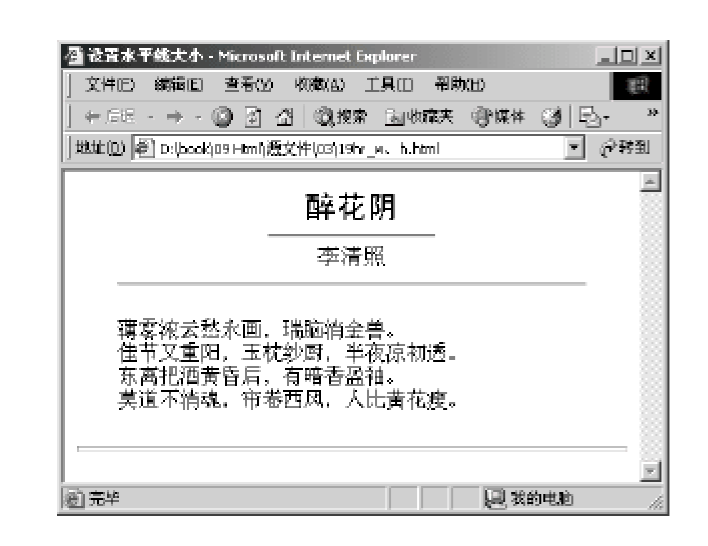


图 3-19　设置水平线宽度和高度

3.4.3　设置水平线的颜色——color

在网页创作时，文字颜色是多姿多彩的，而水平线的色彩也是可以多种多样的。设置了不同色彩的水平线可以为网页增色不少。

语法：<hr color="颜色代码">

说明：颜色代码是十六进制的数值。

实例代码：

```
<html>
<head>
<title>设置水平线的颜色</title>
</head>
<body>
    <center>
        <font face="黑体" size=5 color="#FF0000">天下第一山</font>
    </center>
    <hr width=220 size=3 color="#FF00FF">
    <p>       每一个中国人，都熟悉这座丰碑，但是并不是每一个中国人，都熟悉这座丰碑的基石，来自中国的哪一座山，人民英雄纪念碑的基石，取自泰山山脉的花岗岩，象征着重于泰山。<br>
           每一个中国人都熟悉这座大厦，但是并不是每一个中国人，都熟悉这座大厦的基石，来自中国的哪一座山，人民大会堂的基石就是泰山石，象征着稳如泰山。</p>
    <hr size=5 color="#0000FF">
</body>
</html>
```

运行这段代码，可以看到颜色和大小不同的两条水平线，而这两条水平线将文章的主题映衬得更加醒目，如图 3-20 所示。

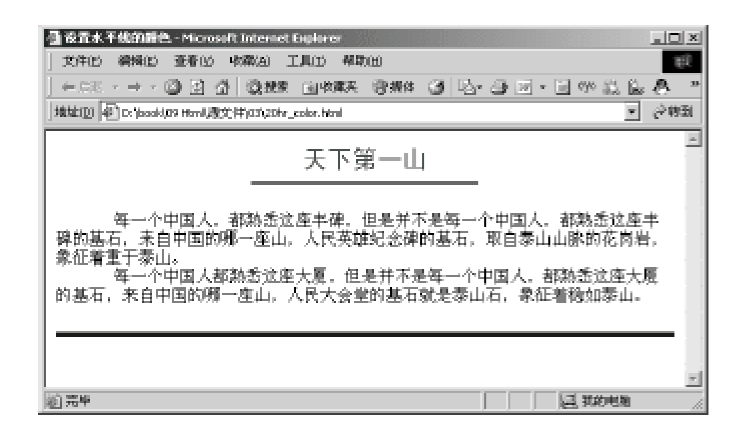


图 3-20　设置水平线的颜色

3.4.4　设置水平线的对齐方式——align

通过前面几个实例可以看到，水平线在默认情况下是居中对齐的。如果希望水平线左对齐或右对齐，就需要使用 align 参数。

语法：<hr align=对齐方式>

说明：在该语法中对齐方式可以有 3 种，包括 left、center 和 right。其中，center 的效果与默认效果相同。

实例代码：

```
<html>
<head>
<title>设置水平线对齐方式</title>
</head>
<body>
	<font face="黑体" size=5 color="#CC0000">醉花阴</font>
	<hr width=130 color="#FF0000" align=left>
	<p align=right><font size=4 color="#990000">李清照</font>
	<hr width=70 size=3 color="#FF0000" align=right></p>
	薄雾浓云愁永画，瑞脑消金兽。<br>
	佳节又重阳，玉枕纱厨，半夜凉初透。<br>
	东离把酒黄昏后，有暗香盈袖。<br>
	莫道不消魂，帘卷西风，人比黄花瘦。
	<hr size=2 color="#FF0000" align=left >
</body>
</html>
```

运行这段代码，可以看到 3 条不同效果的水平线，如图 3-21 所示。

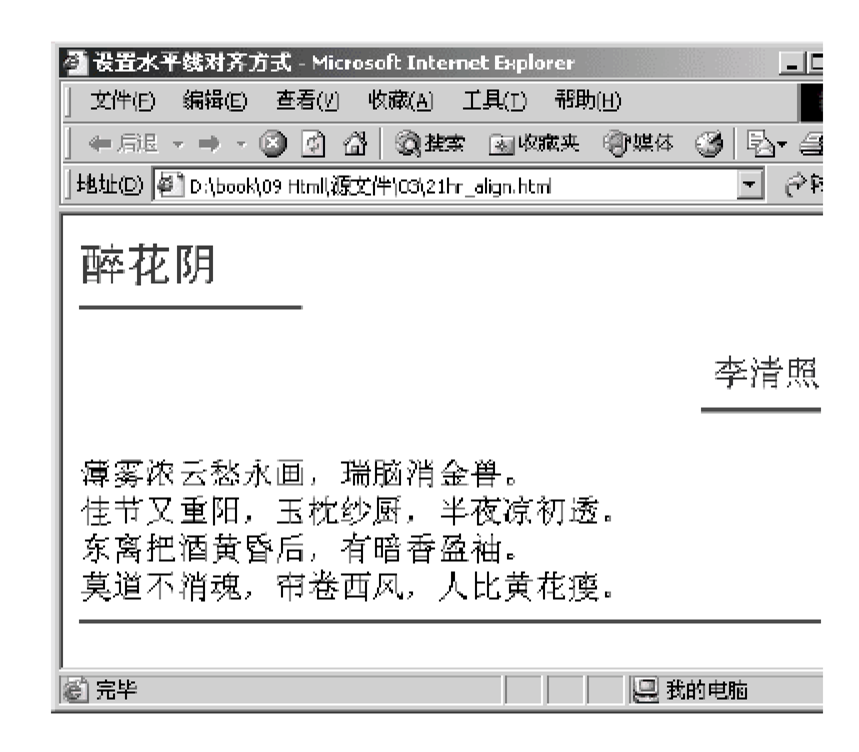


图 3-21　设置水平线的对齐方式

3.4.5　去掉水平线阴影——noshade

在默认情况下，水平线是空心带阴影的立体效果，通过设置 shade 参数可以将水平线的阴影去掉。

语法：<hr noshade>

实例代码：

```
<!--本实例中进行默认水平线和无阴影水平线的对比-->
<html>
<head>
<title>去掉水平线阴影</title>
</head>
<body>
	<center>
		<font face="黑体" size=5 color="#FF0000">泰山</font>
	</center>
	<hr width=130 size=4>
```

```
        <p>   泰山西邻黄河，东依大海，巍峨的雄姿和茂盛的植被，形成了地方性气候，因此生云制雨，是泰山常见的自然现象。</p>
        <hr size=3 noshade>
</body>
</html>
```

运行代码，可以看到如图 3-22 所示的效果。

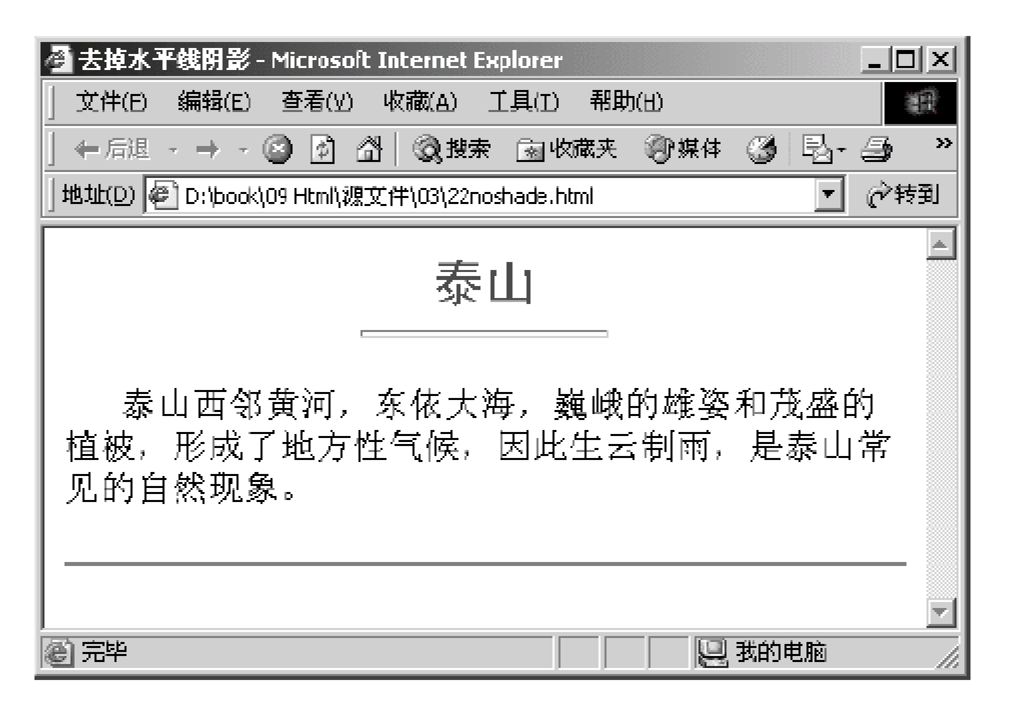


图 3-22　去掉水平线的阴影效果

3.5 其 他 标 记

3.5.1 文字标注标记——ruby

在网页中可以通过添加对文字的标注来说明网页中的某段文字。

语法：

```
<ruby>
    被说明的文字
    <rt>
    文字的标注
    </rt>
</ruby>
```

说明：在这段代码中，被说明的文字就是网页中需要添加标注的那段文字，而文字的标注则是真正的说明文字。

实例代码：

```
<html>
<head>
<title>添加文字标注</title>
</head>
<body>
    <center>
        <ruby>
```

```
        <font face="黑体" size=5 color="#FF0000">天下第一山</font>
        <rt>
        <font color="#660000">泰山</font>
        </rt>
        </ruby>
    </center>
    <p>    泰山周围先后发现了新石器时代八千五百年前的后李文化，七千年前的北辛文化，六千
年左右的大汶口文化，四千年左右的龙山文化和三千年左右的岳石文化。星罗棋布的古文化遗址，就像群星拱北
斗一样，在泰山周围绕了一个巨大的海岱文化圈。</p>
    <hr size=3 noshade>
</body>
</html>
```

运行这段代码，可以在文章的标题上面看到标注文字“泰山”，如图 3-23 所示。在默认情况下，标注文字很小，但是在 HTML 中也可以像设置其他文字一样调整标注文字的各种属性，包括大小、颜色等。

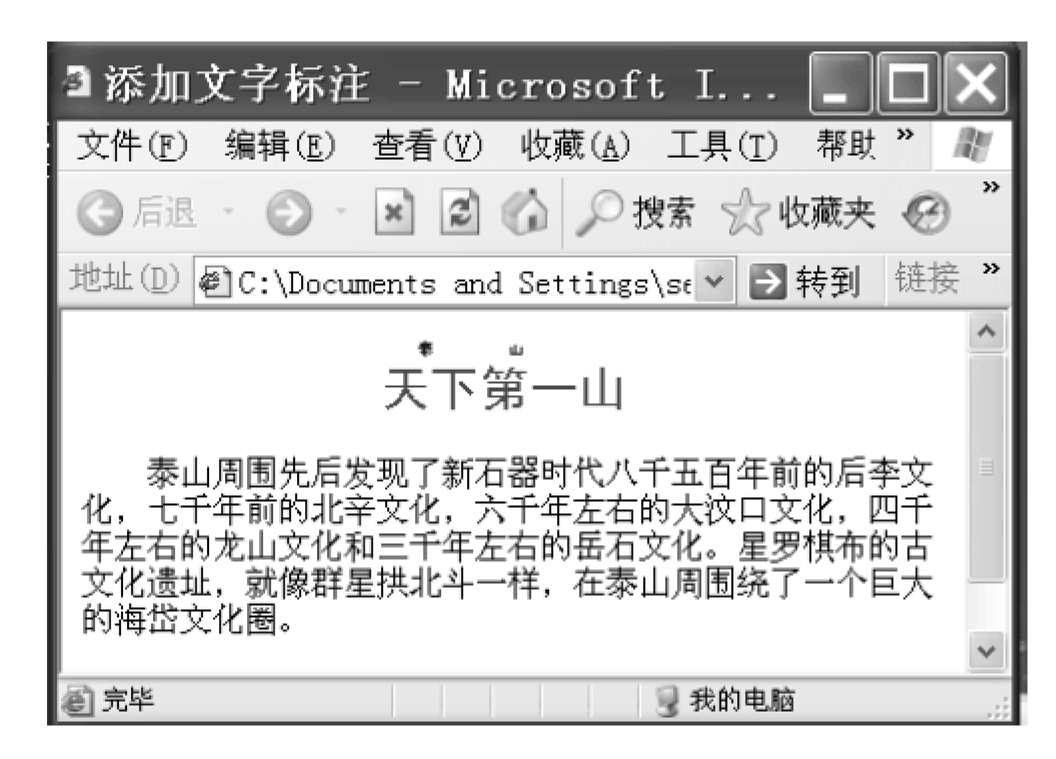


图 3-23　添加标注文字

3.5.2　声明变量标记——var

在使用网页讲解某些知识时，为了统一地突出变量，常常将其设置为斜体。而在 HTML 中也提供了一种标记，用于专门设置变量的效果。

语法：<var>变量</var>

说明：在标记之间的文字就以声明变量的效果显示。

实例代码：

```
<html>
<head>
<title>添加文字标注</title>
</head>
<body>
    <p>定义变量就是给变量赋值。定义变量的格式为：<br>
    变量名:=数值或者表达式的值<br>
    其中符号“:=”是定义符，或者称为赋值符。<br>
    <p>例如定义变量 <var>y</var> 的值为<var> x + 5 </var>，可以表示为：
    <p>       <var>y</var> := <var>x</var> + 5
```

```
</body>
</html>
```

运行这段代码，可以看到如图 3-24 所示的效果。

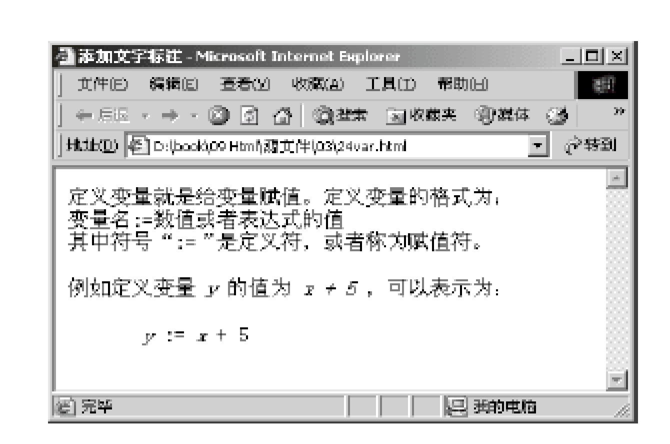


图 3-24　声明变量

3.5.3　忽视 HTML 标签标记——plaintext、xmp

忽视 HTML 标签标记主要是用来使 HTML 标签失去作用，而直接显示在页面中。这一标记在实际中应用并不多。

语法：<plaintext>或<xmp>

说明：这两个标记中的任何一个如果加入到 HTML 代码中，都会使 HTML 标记失去作用，一般放置在<body>标记之后。

实例代码：

```
<html>
<head>
<title>忽视 HTML 标签标记</title>
</head>
<body>
    <plaintext>
    <!--当浏览器宽度不够的时候，文本内容会自动换行显示-->
    <p>HTML 的英文全称是 Hyper Text Markup Language，直译为超文本标记语言，它是全球广域网上描述网
页内容和外观的标准。</p>
    <!--下面这段文字不会自动换行显示，当浏览器宽度不够的时候，会出现滚动条-->
    <p><nobr>HTML 的英文全称是 Hyper Text Markup Language，直译为超文本标记语言，它是全球广域网上
描述网页内容和外观的标准。</nobr></p>
</body>
</html>
```

运行程序的效果如图 3-25 所示。

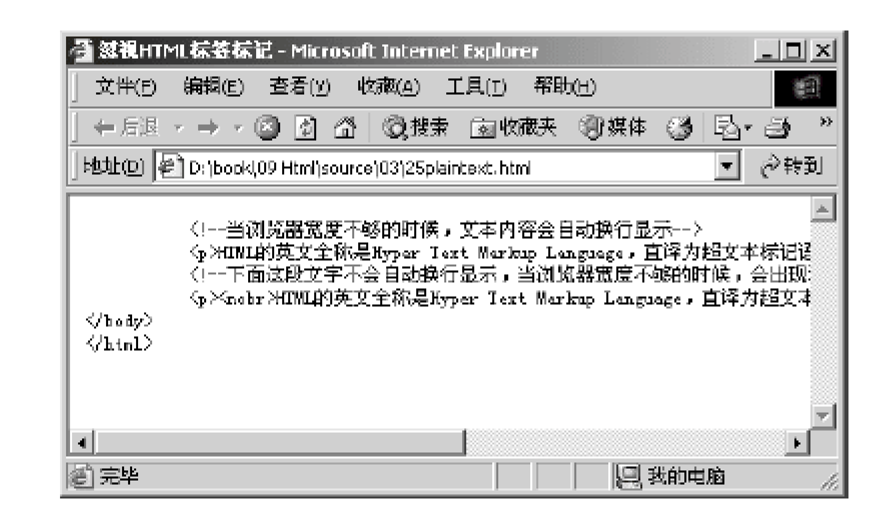


图 3-25　忽略 HTML 标签的作用

第 4 章 列　　表

- 无序列表的设计
- 有序列表的设计
- 定义列表标记——dl
- 菜单列表标记——menu
- 目录列表——dir
- 列表的高级应用
- 列表的嵌套

列表（List）是一种非常实用的数据排列方式，它以条列式的模式来显示数据，使读者能够一目了然。在 HTML 中有 3 种列表，分别是无序列表（Unordered Lists）、有序列表（Ordered Lists）和定义列表（Definition Lists）。

4.1　无序列表的设计

4.1.1　无序列表标记——ul

无序列表的特征在于提供一种不编号的列表方式，而在每一个项目文字之前，以符号作为分项标识。

语法：

```
<ul>
    <LI>第 1 项
    <LI>第 2 项
    <LI>第 3 项
        ……
</ul>
```

说明：在该语法中，使用
标记表示这一个无序列表的开始和结束，而<LI>则表示这是一个列表项的开始。在一个无序列表中可以包含多个列表项。

实例代码：

```
<html>
<head>
<title>创建无序列表</title>
</head>
<body>
    <font size=5 color="#990000">提供下载的软件类别：</font><br><br>
    <ul>
        <LI>系统程序
        <LI>媒体工具
        <LI>管理软件
        <LI>游戏娱乐
    </ul>
</body>
</html>
```

运行这段代码，可以看到窗口中建立了一个无序列表，该列表共包含了 4 个列表项，如图 4-1 所示。

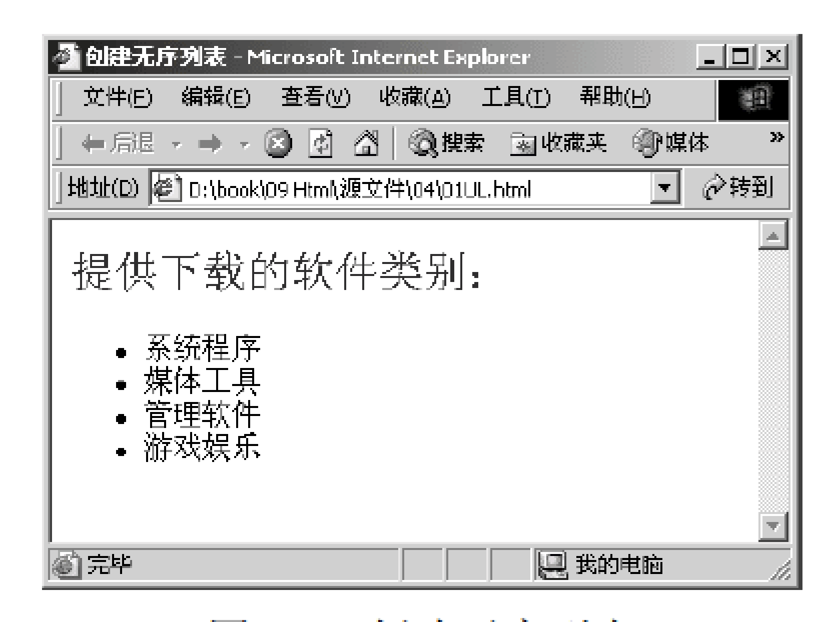


图 4-1　创建无序列表

4.1.2　设置无序列表的符号类型——type

默认情况下，无序列表的项目符号是●，而通过 type 参数可以调整无序列表的项目符号，避免列表符号的单调。

语法：

```
<ul type=符号类型>
    <LI>第 1 项
    <LI>第 2 项
    <LI>第 3 项
        ……
</ul>
```

说明：在该语法中，无序列表其他的属性不变，type 属性则决定了列表项开始的符号。它可以设置的值有 3 个，见表 4-1。其中 disc 是默认的属性值。

表4-1　无序列表的符号类型

类　型　值	列表项目的符号
disc	●
circle	○
square	■

实例代码：

```
<html>
<head>
<title>创建无序列表</title>
</head>
<body>
    <font size=5 color="#990000">出售的图书种类：</font><br><br>
    <ul type=circle>
        <LI>计算机类书籍
        <LI>休闲杂志类书籍
        <LI>考试教材类书籍
        <LI>社会科学类书籍
        <LI>外语原版书
    </ul>
    <hr color="#CC0000" size=2>
    <font size=5 color="#990000">出售的数码产品种类：</font><br><br>
    <ul type=square>
        <LI>电脑配件
        <LI>数码相机
        <LI>手机
        <LI>MP3
    </ul>
</body>
</html>
```

运行这段代码，可以看到除了默认的列表项符号之外，显示了另外两种列表项目符号的效果，如图 4-2 所示。

无序列表的类型定义也可以在<LI>项中，其语法是<LI type=符号类型>，这样定义的结果是对单个项目进行定义，实例代码如下：

```
<html>
<head>
<title>不同的项目符号</title>
</head>
<body>
    <font size=5 color="#990000">出售的图书种类：</font><br><br>
    <ul>
        <LI type=disc >计算机类书籍
        <LI type=circle>休闲杂志类书籍
        <LI type=square >社会科学类书籍
    </ul>
</body>
</html>
```

运行这段代码，效果如图 4-3 所示。

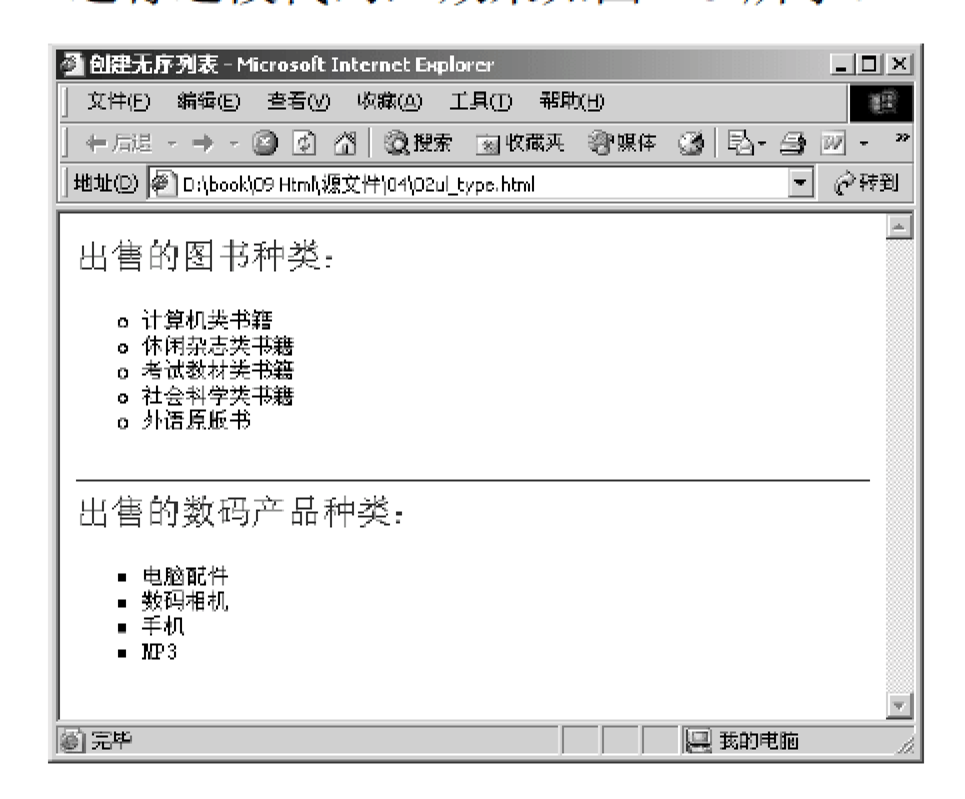


图 4-2　设置无序列表项目符号

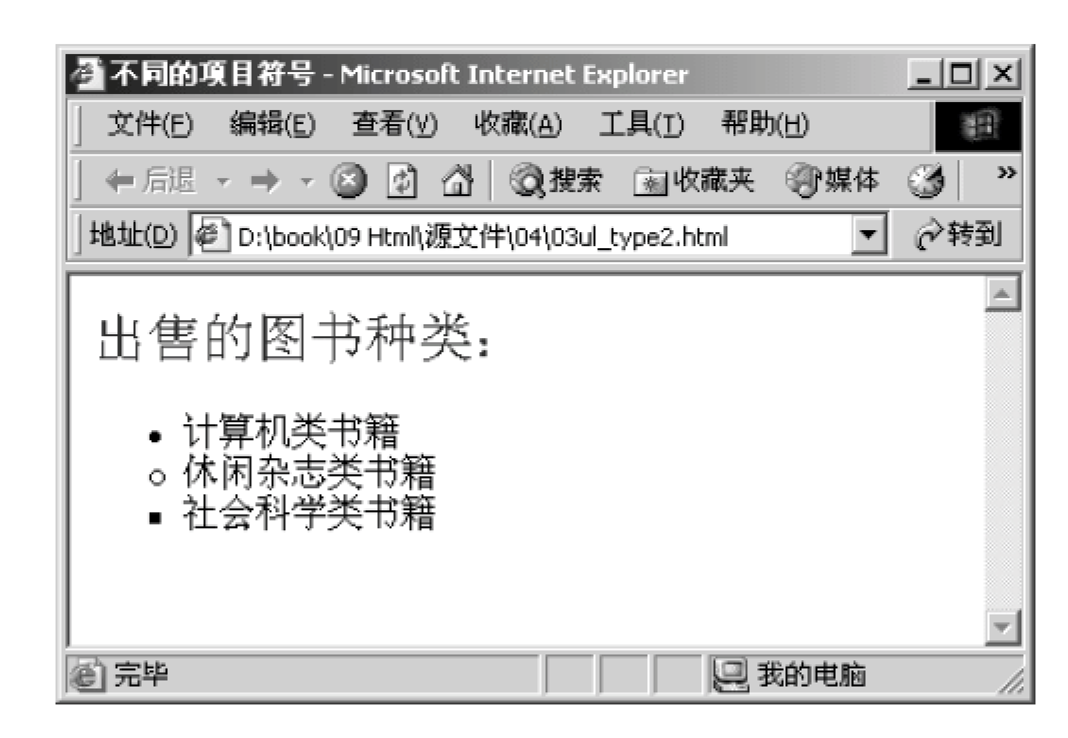


图 4-3　设置不同的项目符号

4.2　有序列表的设计

4.2.1　有序列表标记——ol

有序列表中，各个列表项使用编号而不是符号来进行排列。列表中的项目通常都有先后顺序性，一般采用数字或者字母作为顺序号。

语法：

```
<ol>
    <LI>第 1 项
```

```
        <LI>第 2 项
        <LI>第 3 项
            ……
    </ol>
```

说明：在该语法中，
和
标记标志着有序列表的开始和结束，而<LI>标记表示这是一个列表项的开始，默认情况下，采用数字序号进行排列。

实例代码：

```
<html>
<head>
<title>创建有序列表</title>
</head>
<body>
    <font size=5 color="#990000">创建 HTML 文件的步骤：</font><br><br>
    <ol>
        <LI>启动编写 HTML 文件的软件，如 Dreamweaver
        <LI>编写文件代码
        <LI>保存文件
        <LI>运行文件并查看效果
    </ol>
</body>
</html>
```

运行这段代码，可以看到序列前面包含了顺序号，如图 4-4 所示。

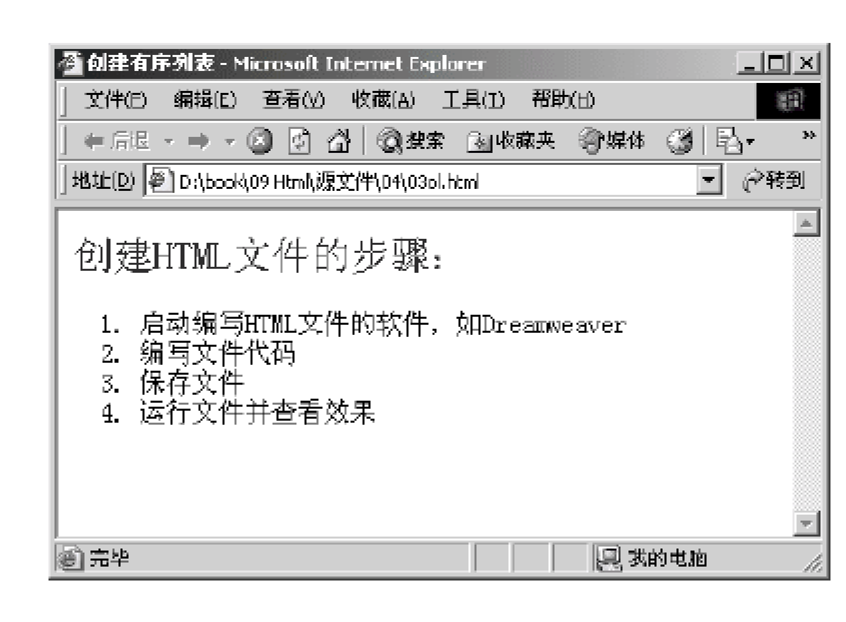


图 4-4　有序列表

4.2.2　有序列表的序号类型——type

默认情况下，有序列表的序号是数字的，通过 type 属性可以调整序号的类型，例如将其修改成字母等。

语法：

```
    <ol type=序号类型>
        <LI>第 1 项
        <LI>第 2 项
        <LI>第 3 项
            ……
    </ol>
```

说明：在该语法中，序号类型可以有 5 种，见表 4-2。

表4-2　有序列表的序号类型

type取值	列表项目的序号类型
1	数字1，2，3，4……
a	小写英文字母a，b，c，d……

续表

type取值	列表项目的序号类型
A	大写英文字母A，B，C，D……
i	小写罗马数字i，ii，iii，iv……
I	大写罗马数字I，II，III，IV……

实例代码：

```
<html>
<head>
<title>创建有序列表</title>
</head>
<body>
    <font size=4 color="#990000">小小蚂蚁测试你如何面对人际关系</font><br><br>
    你和朋友约好一起逛街，可是你的朋友迟到了。于是你就开始在地上乱画。突然，你发现有很多蚂蚁正在成群结队地往前行进。你认为这些蚂蚁正在做什么？
    <ol type=A>
        <LI>正在搬家
        <LI>大家正要前往救助掉落洞穴中的伙伴
        <LI>发现好吃的食物，大家正要去搬运
        <LI>正要去袭击侵入它们势力范围的敌人
    </ol>
    <hr size=2 color="#CC0000">
    <font size=4 color="#990000">走路姿势透露你的另一面</font><br><br>
    走路是我们每天都要做的事情，似乎就像我们呼吸空气一样平常，走路会藏有什么玄机呢？其实，这里面的奥妙可大着呢！每个人有每个人的习惯，由此，我们每个人走路的姿态就会有所不同，就在这不同的姿势中，你的性格特征以及爱情玄机就一点一滴地泄露出来了哟！
    <ol type=i>
        <LI>走路的时候速度比较快，通常五指伸得笔直
        <LI>走路时的速度一般，手掌常自然地握成拳状
        <LI>常将一只手插进口袋，或两手同时插进口袋
        <LI>走路的时候速度比较慢，五指自然微微弯曲
    </ol>
</body>
</html>
```

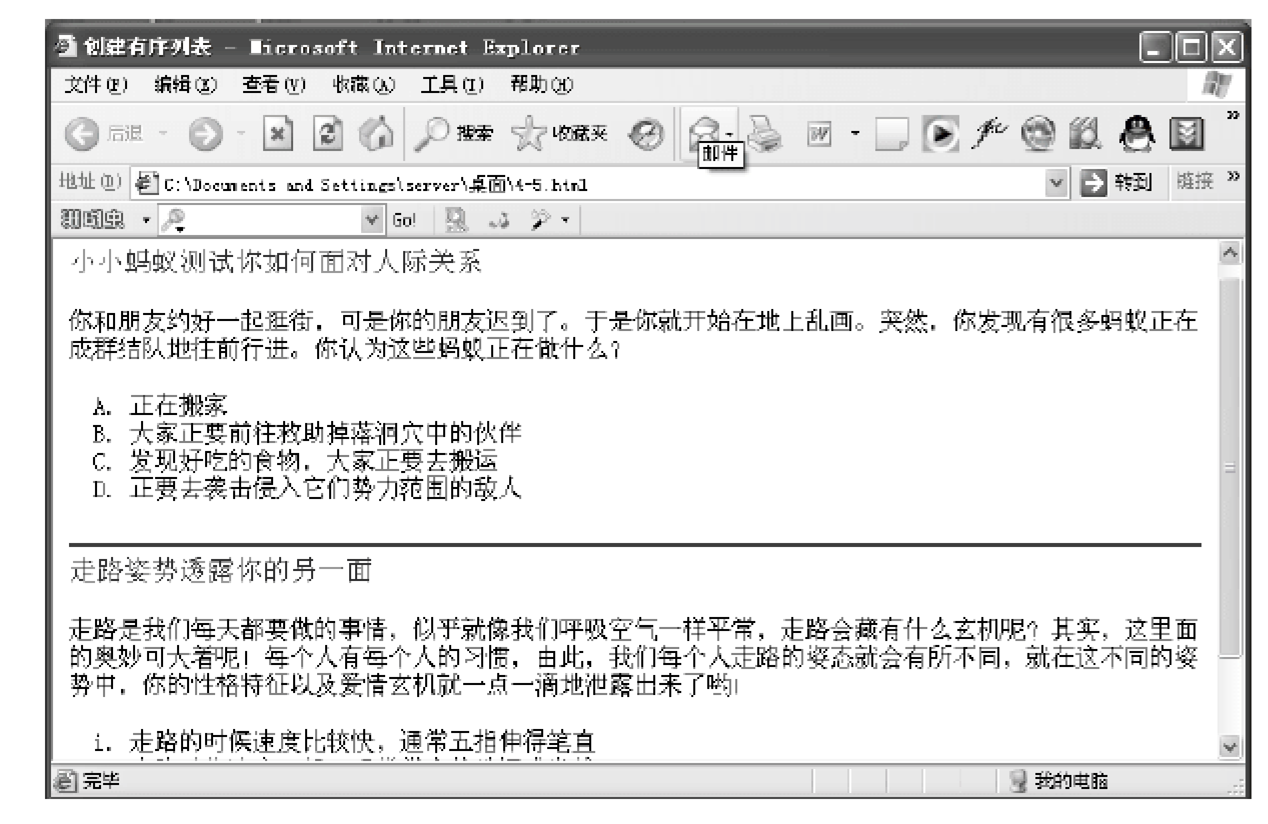


图 4-5　有序列表的类型

运行这段代码，可以实现有序列表的不同类型的序号排列，如图 4-5 所示。

4.2.3　有序列表的起始数值——start

默认情况下，有序列表的列表项是从数字 1 开始的，通过 start 参数可以调整起始数值。这个数值

可以对数字起作用，也可以作用于英文字母或者罗马数字。

语法：

```
<ol start=起始数值>
    <LI>第 1 项
    <LI>第 2 项
    <LI>第 3 项
        ……
</ol>
```

说明：在该语法中，起始数值只能是数字，但是同样可以对字母和罗马数字起作用。

实例代码：

```
<html>
<head>
<title>有序列表的起始值</title>
</head>
<body>
    <font size=4 color="#990000">这些动物都是国家保护动物：</font><br><br>
    <ol start=3>
        <LI>大熊猫
        <LI>金丝猴
        <LI>娃娃鱼
        <LI>变色龙
    </ol>
    <hr size=2 color="#CC0000">
    <font size=4 color="#990000">读完本期故事会，选择您认为最好的一篇文章：</font><br><br>
    <ol type=A start=5>
        <LI>晴天、雨天
        <LI>我长大了
        <LI>风中的百合花
    </ol>
</body>
</html>
```

运行这段代码，效果如图 4-6 所示，其中定义了不同的起始编号。

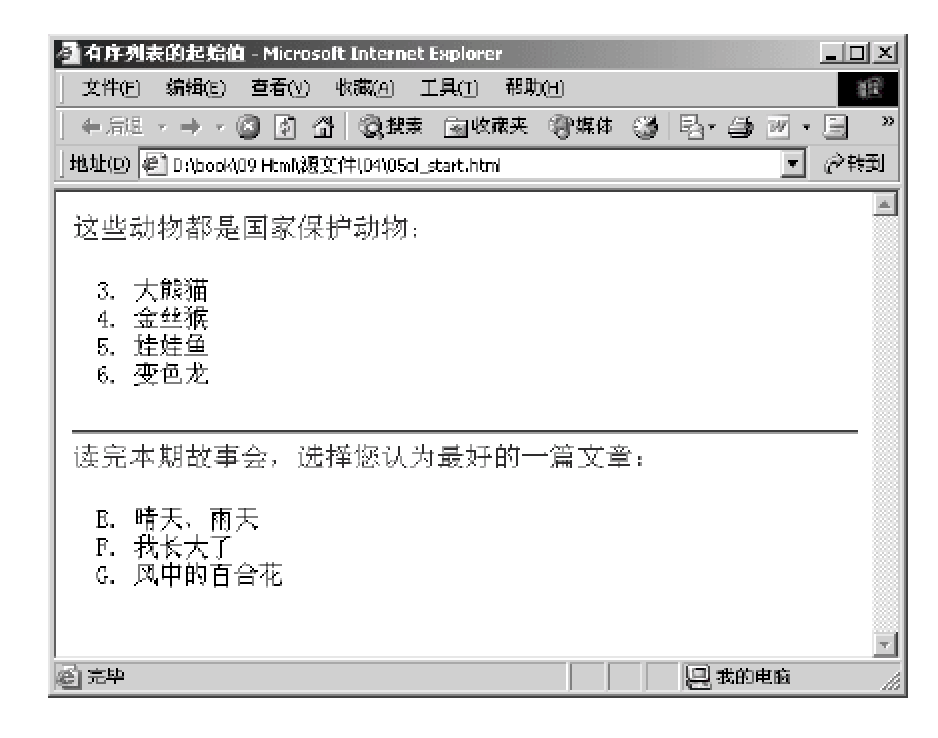


图 4-6　设置有序列表的起始编号

4.3　定义列表标记——dl

在 HTML 中还有一种列表标记，称为定义列表（Definition Lists）。不同于前两种列表，它主要用于解释名词，包含两个层次的列表，第一层次是需要解释的名词，第二层次是具体的解释。

语法：

```
<dl>
    <dt>名词 1<dd>解释 1
    <dt>名词 2<dd>解释 2
    <dt>名词 3<dd>解释 3
        ……
</dl>
```

说明：在该语法中，<dl>标记和</dl>标记分别定义了定义列表的开始和结束，<dt>后面就是要解释的名称，而在<dd>后面则添加该名词的具体解释。作为解释的内容在显示时会自动缩进，有些像字典中的词语解释。

实例代码：

```
<html>
<head>
<title>创建定义列表</title>
</head>
<body>
    <font size=5 color="#000099">网页创作的相关知识：</font><br><br>
    <dl>
        <dt>HTML<dd>HTML 是英文 Hyper Text Markup Language 的缩写，即超文本标志语言
        <dt>CSS<dd>CSS 是 Cascading Style Sheets（层叠样式表单）的简称，一种设计网页样式的工具
        <dt>JavaScript<dd>JavaScript 是一种新的描述语言，可以被嵌入 HTML 的文件之中
        <dt>CGI<dd>CGI 是一段程序，它运行在 Server 上，提供同客户端 HTML 页面的接口
    </dl>
</body>
</html>
```

运行这段代码，可以实现如图 4-7 所示的定义列表效果。

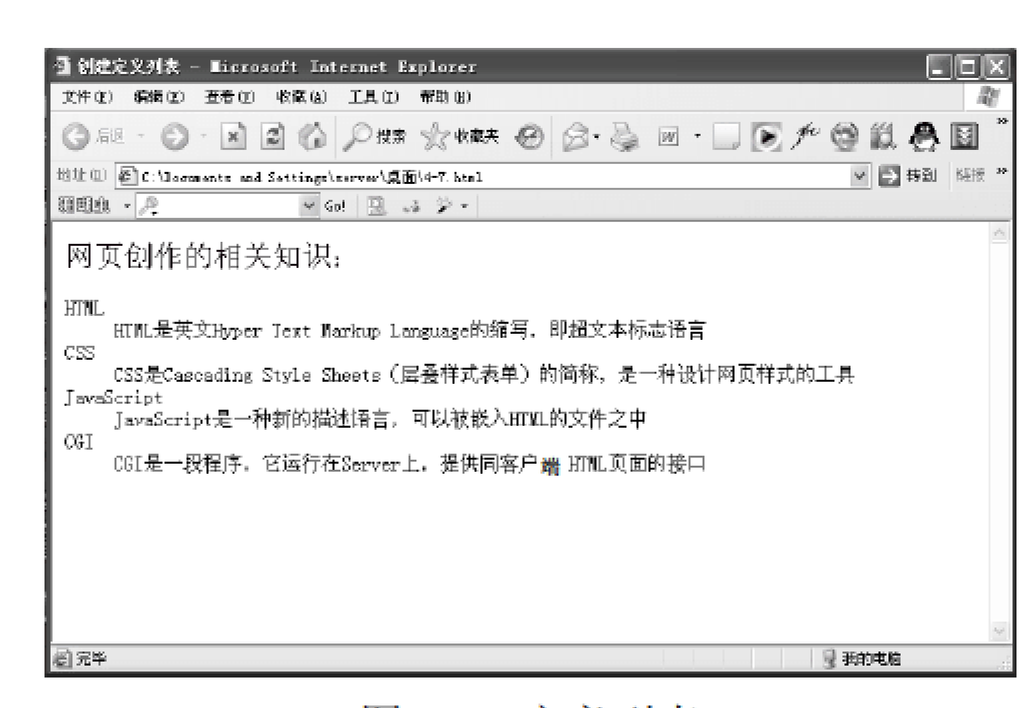


图 4-7　定义列表

4.4　菜单列表标记——menu

菜单列表主要用于设计单列的菜单列表。菜单列表在浏览器中的显示效果和无序列表是相同的，因此它的功能也可以通过无序列表来实现。

语法：

```
<menu>
    <LI>列表项 1
    <LI>列表项 2
    <LI>列表项 3
        ……
</menu>
```

说明：在该语法中，<menu>和</menu>标志着菜单列表的开始和结束。

实例代码：

```
<html>
<head>
<title>创建菜单列表</title>
</head>
<body>
    <font size=5 color="#000066">本章中介绍的列表主要包括：</font><br><br>
    <menu>
        <LI>无序列表
        <LI>有序列表
        <LI>定义列表
        <LI>菜单列表
        <LI>目录列表
    </menu>
</body>
</html>
```

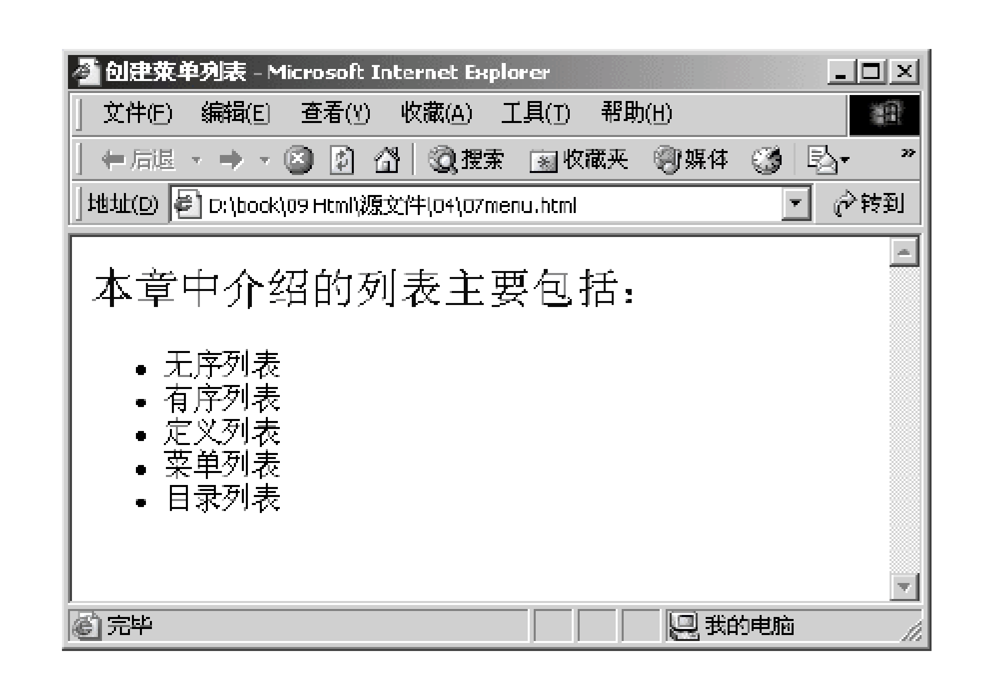


图 4-8　菜单列表的效果

运行这段代码，效果如图 4-8 所示。

4.5　目录列表——dir

目录列表一般用来创建多列的目录列表，它在浏览器中的显示效果与无序列表相同，因此它的功能也可以通过无序列表来实现。

语法：

```
<dir>
    <LI>列表项 1
```

```
        <LI>列表项 2
        <LI>列表项 3
            ……
    </dir>
```

说明：在该语法中，<menu>和</menu>标志着菜单列表的开始和结束。

实例代码：

```
<html>
<head>
<title>创建目录列表</title>
</head>
<body>
    <font size=5 color="#000066">禅</font><br><br>
    <dir>
        <LI>诸恶不作，众善奉行
        <LI>人间万事塞翁马
        <LI>一切众生希有佛性
        <LI>面面佛面行行佛行
    </dir>
</body>
</html>
```

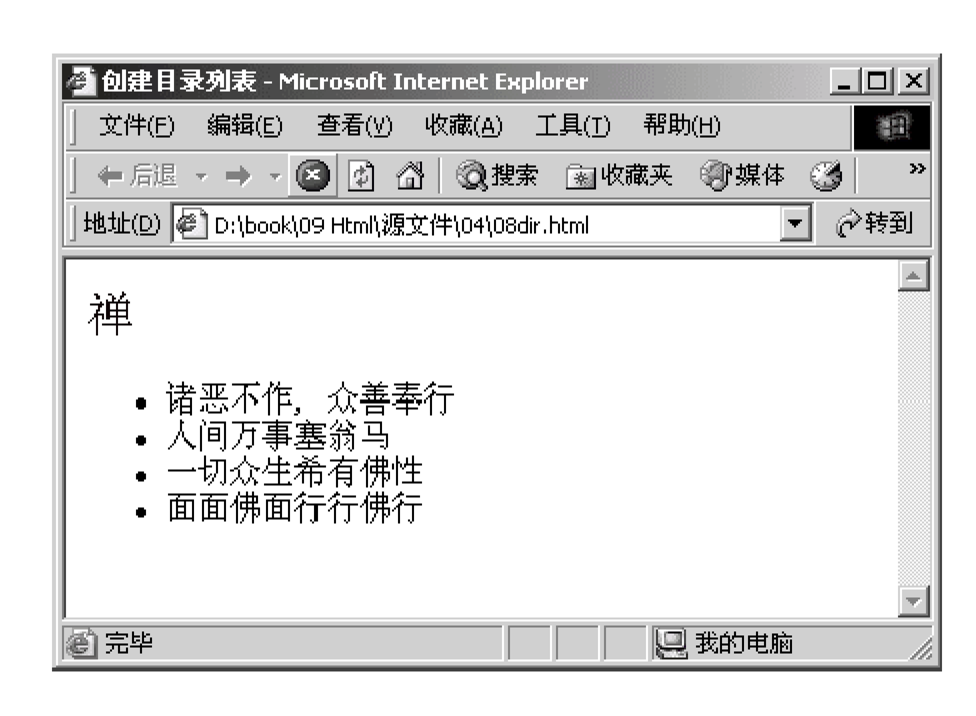


图 4-9　目录列表

运行这段代码，效果如图 4-9 所示。

4.6　列表的高级应用

4.6.1　列表的简化

当制作的列表清单过长时，浏览器可能需要利用滚动条才能看到所有的内容。为了节省空间，避免过多的留白，可以考虑利用 compact 参数来紧密排列各要项的内容。

语法：<dl compact></dl>

说明：在该语法中，compact 参数除了与定义列表结合之外，还可以与各种列表结合使用，达到列表的简化功能。

实例代码：

```
<html>
<head>
<title>创建定义列表</title>
</head>
<body>
    <font size=5 color="#000099">网页创作的相关知识：</font><br><br>
    <dl compact>
        <dt>HTML<dd>HTML 是英文 Hyper Text Markup Language 的缩写，即超文本标志语言
        <dt>CSS<dd>CSS 是 Cascading Style Sheets（层叠样式表单）的简称，是一种设计网页样式的工具
```

```
        <dt>CGI<dd>CGI 是一段程序，它运行在 Server 上，提供同客户端 HTMl 页面的接口
    </dl>
</body>
</html>
```

运行这段代码，与本章 4.3 节的实例相对比，可以看到解释内容没有单独开始一行，而是直接在词语后面，从而节省了空间，如图 4-10 所示。

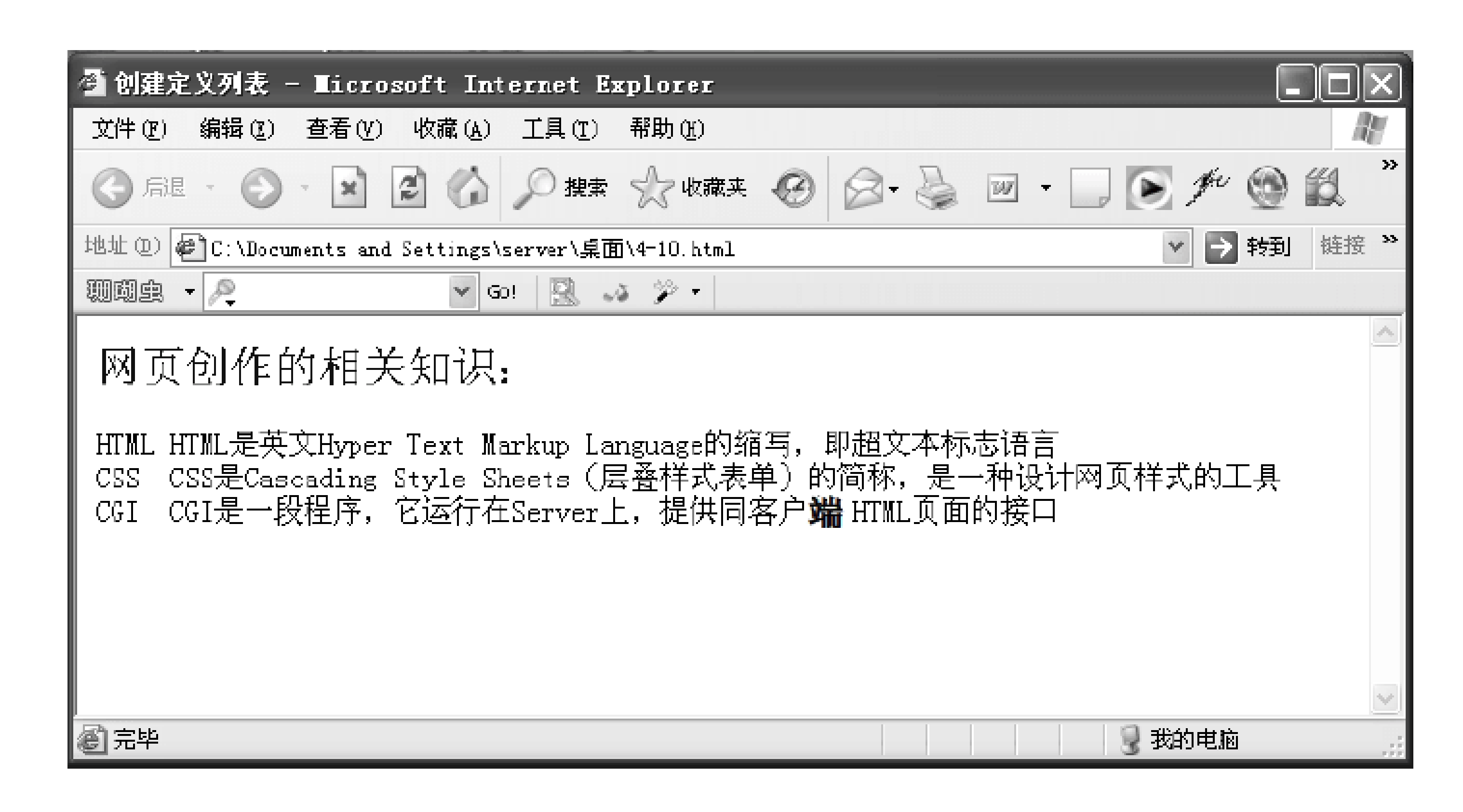


图 4-10　简化列表

4.6.2　设置列表文字的颜色

在创建列表时，可以单独控制列表中文字的颜色。在这里可以直接对文字进行颜色设置。

语法：<LI>列表项的内容<LI>

说明：在该语法中，列表项内容的颜色就变成了设置后的颜色。也可以在列表中进行整体颜色的设置。

实例代码：

```
<html>
<head>
<title>设置列表文字的颜色</title>
</head>
<body>
    <font size=5 color="#990000">文学世界：</font><br><br>
    <ul><font color="#CC0000">
        <LI>散文精选</LI>
        <LI>小说天地</LI>
        <LI>诗词歌赋</LI>
        </font>
    </ul>
    <hr size=2 color="#CC0000">
    <font size=5 color="#000099">散文精选：</font><br><br>
    <ul>
        <LI><font color="#996600">光阴的故事</font></LI>
```

```
            <LI><font color="#3366FF">菁菁校园情</font></LI>
            <LI><font color="#CC0099">漫漫旅途路</font></LI>
      </ul>
</body>
</html>
```

运行这段代码，效果如图 4-11 所示。

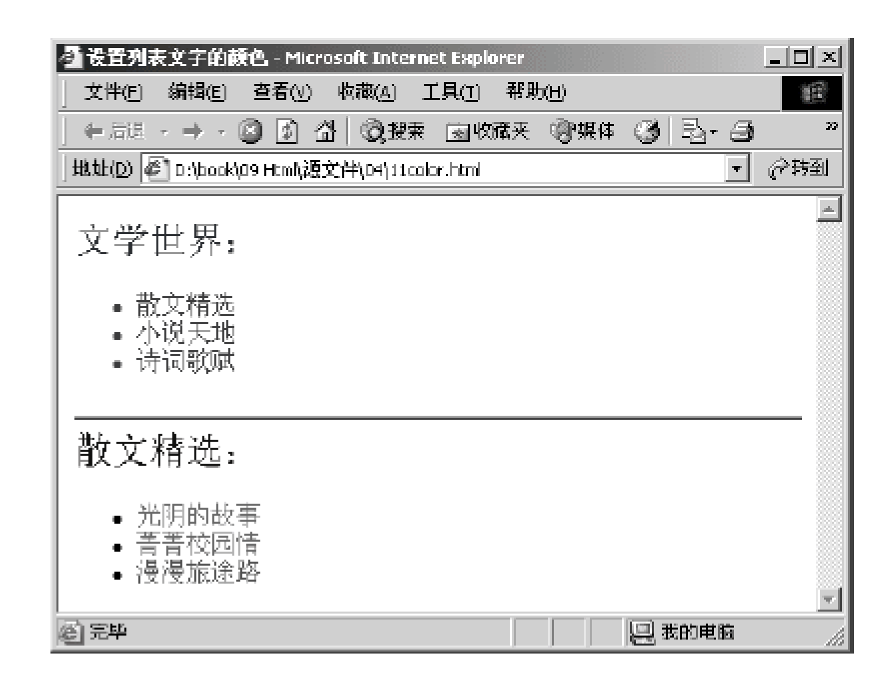


图 4-11　设置列表文字的颜色

4.7　列表的嵌套

最常见的列表嵌套模式就是有序列表和无序列表的嵌套，可以重复地使用<ol>和<ul>标记组合实现。实例代码：

```
<html>
<head>
<title>创建有序列表</title>
</head>
<body>
      <ul type=square>
      <LI><font size=5 color="#990000">探测你的恋爱冒险心：</font><br>
            假设你想要让喜欢的他（她）有脸红心跳的感觉，所以打算穿一件白色的裙子（裤子），不过上半身却不知道要配什么才好，于是你跑到商场买了一件今年最流行的条纹 T-shirt，你会选择以下哪种款式的上衣呢？
            <br><br>
            <ol type=A>
                  <LI>运动衫
                  <LI>套头衫
                  <LI>V 领休闲衫
                  <LI>带帽衫
            </ol>
      </LI><br>
      <LI><font size=5 color="#990000">选鬼屋看你适合什么工作：</font><br>
            你在旅行时不小心在荒山野岭迷了路，这时天色已晚，你发觉附近只有一间小屋子，逼不得已只好向主人借宿，可是屋主老夫妇却告诉你屋子的四个房间都闹鬼，一定要住下来的情况下，你会选择哪个房间呢？
            <br><br>
            <ol type=A >
```

```
                <LI>有个人头从窗外恶狠狠瞪着你睡觉的房间
                <LI>厕所会传来开关门声和女人叹息声的房间
                <LI>你一躺上去床就开始摇晃不让你睡的房间
                <LI>半夜醒来看到一个无头鬼坐在床边的房间
            </ol>
        </LI>
        </ul>
</body>
</html>
```

运行这段代码，效果如图 4-12 所示。

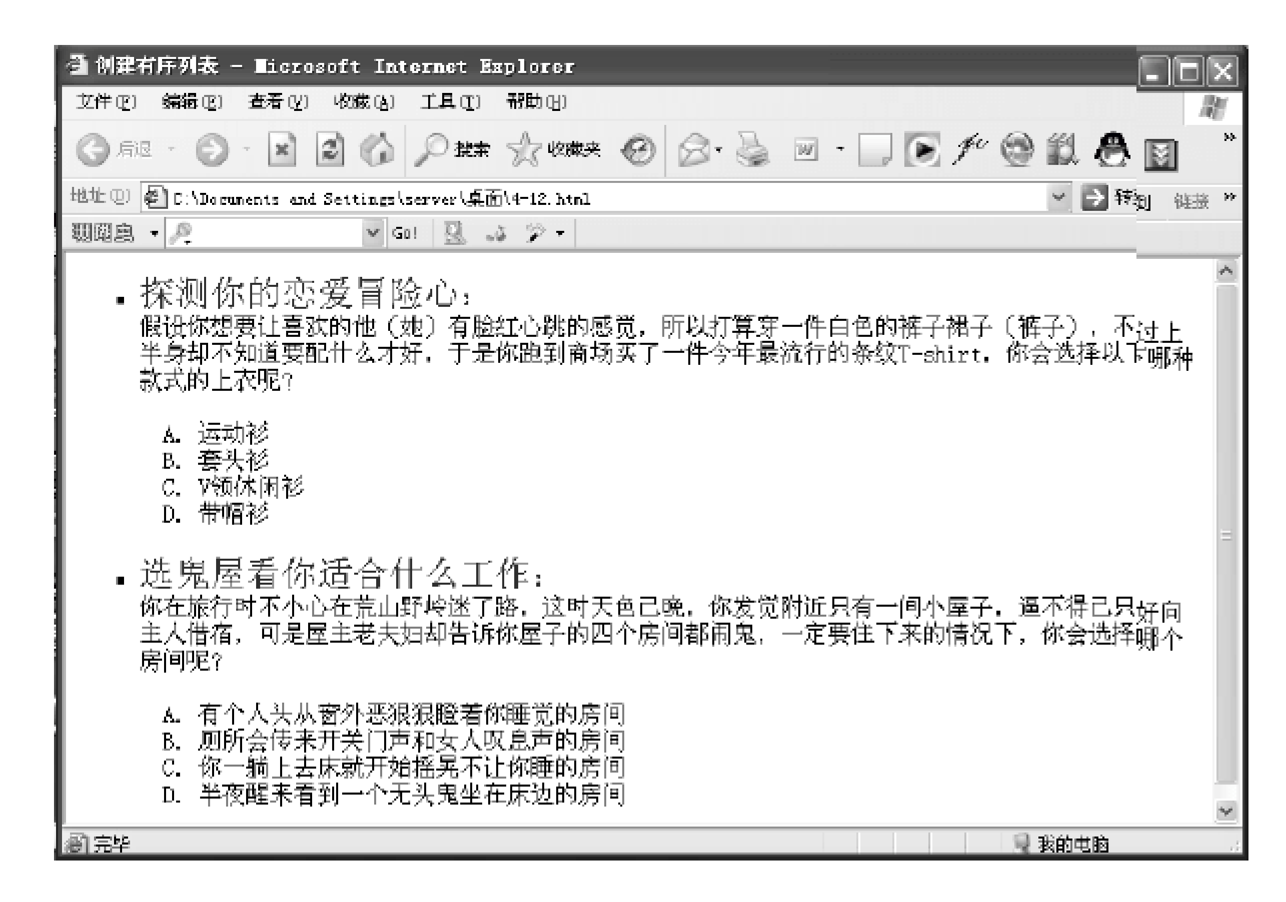


图 4-12　列表的嵌套

第5章

超　链　接

- 超链接基本知识
- 超链接的建立
- 内部链接
- 书签链接
- 外部链接

所谓超链接就是当单击某个字或某个图片时，就可以打开另外一个画面。它的作用对网页来说相当重要，如果没有它，那便只能在每打开一个页面时就要在地址栏内输入一次地址了。

5.1　超链接基本知识

5.1.1　超链接

对于初次接触网页设计的读者来说，可能对于超链接的概念还不是很明白。超链接就是从一个网页转到另一个网页的途径。

超链接是网页的重要组成部分。如果说文字、图片是网站的躯体，那么超链接就是整个网站的神经细胞，它把整个网站的信息有机地结合到一起。链接能使浏览者从一个页面跳转到另一个页面，实现文档互联、网站互联。

超文本链接（hypertextlink）通常简称为超链接（hyperlink），或者简称为链接（link）。链接是 HTML 的一个最强大和最有价值的功能。链接是指文档中的文字或者图像与另一个文档、文档的一部分或者一幅图像链接在一起。

5.1.2　绝对路径

绝对路径就是主页上的文件或目录在硬盘上的真正路径。使用绝对路径定位链接目标文件比较清晰，但是有两个缺点：一是需要输入更多的内容，二是如果该文件被移动了，就需要重新设置所有的相关链接。例如在本地测试网页时链接全部可用，但是到了网上就不可用了。这就是路径设置的问题。例如设置路径为“C:\Program files\1.htm”，在本地确实可以找到，但是到了网站上该文件便不一定在这个路径下了，所以就会出问题。

5.1.3　相对路径

首先分析一下为什么会发生图片不能正常显示的情况。举一个例子，现在有一个页面 index.htm，在这个页面中链接有一张图片 photo.jpg，它们的绝对路径如下：

c:\website\index.htm

c:\website\img\photo.jpg

如果使用绝对路径 c:\website\img\photo.jpg，那么在自己的计算机上将一切正常。因为确实可以在指定的位置（即 c:\website\img\photo.jpg）上找到 photo.jpg 文件。但是当将页面上传到网站时就很可能会出错了，因为网站可能在服务器的 c 盘，也可能在 d 盘，也可能在 aa 目录下，更可能在 bb 目录下，也就是说很有可能不存在 c:\website\img\photo.jpg 这样一个路径。此时就用到相对路径。所谓相对路径，顾名思义就是自己相对于目标位置。在上例的 index.htm 中链接的 photo.jpg 可以使用 img\photo.jpg 来定位文件，那么不论将这些文件放到哪里，只要它们的相对关系没有变，就不会出错。

在编程中使用“..\”来表示上一级目录，“..\..\”表示上上级的目录，以此类推。再看几个例子，注意所有例子中都是 index.htm 文件中链接有一张图片 photo.jpg。

看下面这个例子：

c:\website\web\index.htm

c:\website\img\photo.jpg

在此例中，index.htm 中链接的 photo.jpg 应该怎样表示呢？

错误写法：img\photo.jpg，这种写法是不正确的。在此例中，对于 index.htm 文件来说，img\photo.jpg 所代表的绝对路径是 c:\website\web\img\photo.jpg，显然不符合要求。

正确写法：使用..\img\photo.jpg 的相对路径来定位文件。

总结一下，对于相对路径来说，设置起来一般有如下 3 种写法：

- 同一目录下的文件：只需要输入链接文件的名称即可，如01.html。
- 上一级目录中的文件，在目录名和文件名之前加入“../”，如../04/02.html；如果是上两级，则需要加入两个“../”，如../../file/01.html。
- 下一级目录：输入目录名和文件名，之间以“/”隔开，如：Html/05/01.html。

除了绝对路径和相对路径之外，还有一种称为根目录。根目录常常在大规模站点需要放置在几个服务器上，或者一个服务器上同时放置多个站点时使用。其书写形式很简单，只需要以“/”开始，表示根目录，之后是文件所在的目录名和文件名。

5.2 超链接的建立

5.2.1 超链接标记的基本语法

超级链接的语法根据其链接对象的不同而有所变化，但都是基于<A>标记的。

语法：<A href="文件名">链接元素</A>或<A href="URL">链接元素</A>

说明：在该语法中，链接元素可以是文字，也可以是图片或其他页面元素。其中 href 是 hypertext reference 的缩写。通过超级链接的方式可以使各个网页之间链接起来，使网站中众多的页面构成一个有机整体，使访问者能够在各个页面之间跳转。超级链接可以是一段文本、一幅图像或其他网页元素，当在浏览器中用鼠标单击这些对象时，浏览器可以根据指示载入一个新的页面或者转到页面的其他位置。

下面具体讲解各种超链接的创建方法。

5.2.2 建立文本超链接

在网页中，文本超链接是最常见的一种。它通过网页中的文件和其他的文件进行链接。

语法：<A href="链接地址">链接文字</A>

说明：在该语法中，链接地址可以是绝对地址，也可以是相对地址。

实例代码：

```
<html>
<head>
<title>文本链接</title>
</head>
<body>
```

```
    <center><h3>神话故事：八仙斗花龙</h3></center>
    <a href="next.html">下一篇：女娲补天</a><br><br>
        传说，有一天八仙要到东海去游蓬莱岛。本来，腾云驾雾，一眨眼就可到，可是吕纯阳偏偏别出心裁，提出要乘船过海，观赏海景。他拿来铁拐李的拐杖，往海里一抛，喝声“变”，顿时变成一艘宽敞、漂亮的大龙船，八位大仙坐船观景，喝酒斗歌，好不热闹。不料，因此惹出一场麻烦来。原来，龙宫里有条花鳞恶龙，是龙王的第七个儿子，称为“花龙太子”。这天，他闲得没事，在水晶宫外游荡，忽闻海面上有仙乐之声，便循声寻去，猛见一条雕花龙船，内坐八位奇形怪状的大仙，其中有个妙龄女郎，楚楚动人，花龙太子见此仙姿，魂魄俱消，早忘了师傅南极仙翁的忠告，忘了龙王母的训导，想入非非，似魔似痴地迷上何仙姑了。<br>
        八仙在海上寻欢作乐，怎会想到花龙太子半路挡道。平静的海面突然掀起一个浪头，将雕花龙船打翻了。张果老眼尖，翻身爬上毛驴背；曹国舅心细，脚踏巧板浪里漂；韩湘子放下仙笛当坐骑；汉钟离打开蒲扇垫脚底；蓝采和攀住了花篮边；铁拐李失了拐杖，幸亏抱着个葫芦；只有吕纯阳，毫无戒备，弄了个浑身湿透。这时，汉钟离慌忙检点人数。点过来，点过去，只有七位大仙。男的俱在，独缺一个何仙姑。奇怪，这何仙姑到哪里去了呢？汉钟离掐指一算，大吃一惊，原来是花龙太子拦路抢亲，把何仙姑抢到龙宫里去了。这一回，大仙们可大动肝火了。个个咬牙切齿，杀气腾腾，直奔龙宫。花龙太子知道七仙不会善罢干休，早在半路上伺候着。他见大仙们来势凶猛，慌忙挥舞珍珠鳌鱼旗，催动虾兵蟹将，掀起漫海大潮，向七仙淹来。汉钟离挺着大肚子，飘飘然降落潮头，轻轻扇动蒲扇。只听“呜…忽…”一声，一阵狂风把万丈高的虾兵蟹将都扇到九霄云外去了，吓得四大天王连忙关了南天门。花龙太子见汉钟离破了他的阵势，忙把脸一抹，喝声“变”。海里突然窜出一條巨鲸，张开闸门似的大口来吞汉钟离。汉钟离急忙扇动蒲扇，不料那巨鲸毫无惧色，嘴巴越张越大。这下，汉钟离可慌了神了……
</body>
</html>
```

在这段代码中，有一个文本链接“下一篇：女娲补天”，它链接到了当前目录下的 next.html 文件。文件 next.html 的实例代码如下：

```
<html>
<head>
<title>文本链接</title>
</head>
<body>
    <center><h3>神话故事：女娲补天</h3></center>
    <a href="first.html">上一篇：八仙斗花龙</a><br><br>
        有一天，大龙和精卫、小太极一起到远古时代去玩，居然看到了传说中能顶住上天、撑开天与地的「不周山」，也遇到了美丽的女娲娘娘，甚至看到女娲补天的过程……那真是一场惊天地而泣鬼神的经历，讲来都令人鼻酸！<br>
        天上的大神女娲，用泥土做成泥娃娃，再赋予它们生命。从此女娲创造了人，人们世世代代繁衍生息，过着幸福的生活。 然而，好景不长。有一年，火神祝融和水神共工打起仗来。共工把撑天的柱子不周山撞倒了，天塌下半边来，砸了很多窟窿，把地也砸裂了，地上的洪水及天河的水不停地漏下来，造成大地上水患、火海，真是悲惨啊！<br>
        女娲为了解救人类，决定采石补天，用五色石把天补起来，再用东海神龟的四只脚顶住苍天。 眼看着补天的大功就要告成，却发现五色石不够用，大洞还没补好，这可怎么办呢？慈悲的女娲只好牺牲自己的生命，用身体来补天上的大洞，亲眼看到那场过程，大龙、精卫和小太极，以及所有的泥娃娃们都哭红了眼睛！经过几次努力，女娲终于补好了天，天地间恢复了宁静，还出现了五彩云霞。一切生物又都生机勃勃地活在大地上。女娲为人类和世界万物的生存而献身，真是令人感动。
</body>
</html>
```

运行文件 first.html，可以看到文件中包含一个文本链接“下一篇：女娲补天”，其效果如图 5-1 所示。

单击页面中的文本“下一篇：女娲补天”，可以将页面转到 next.html 文件，如图 5-2 所示。

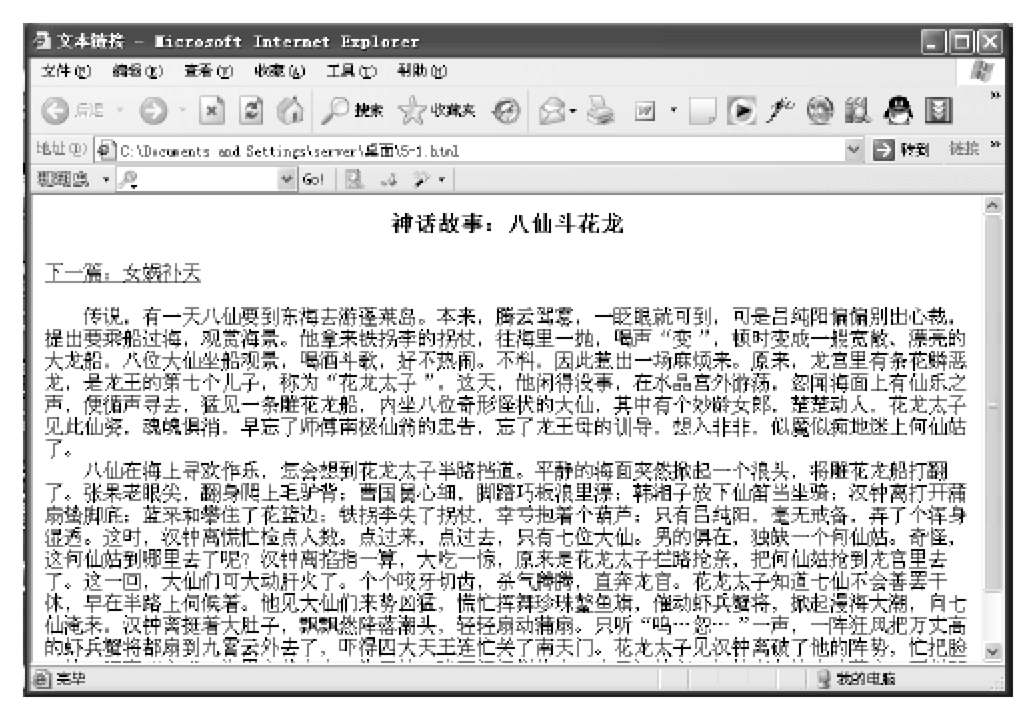


图 5-1　文本链接的页面效果

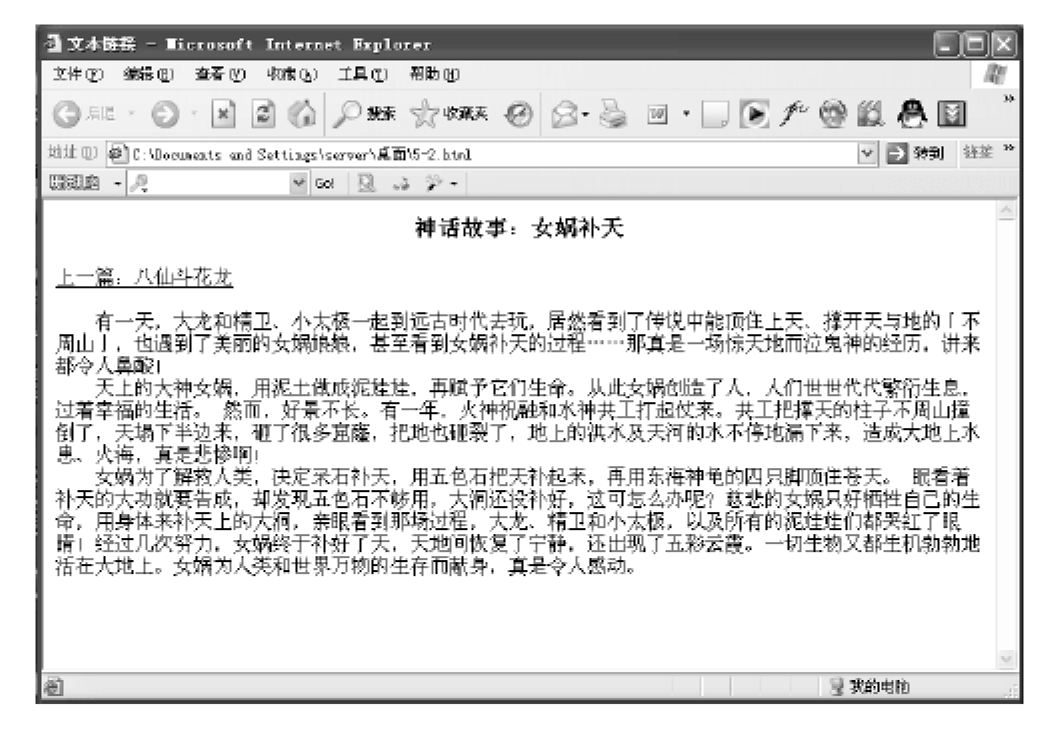


图 5-2　打开的链接页面

5.2.3　设置超链接的目标窗口

在创建网页的过程中，有时候并不希望超链接的目标窗口将原来的窗口覆盖，比如希望不论链接到何处，主页面都保留在原处。这时可以通过 target 参数设置目标窗口的属性。

语法：<A href="链接地址" target=目标窗口的打开方式>链接元素</A>

说明：在该语法中，target 参数的取值有 4 种，见表 5-1。

表5-1　目标窗口的设置

target值	目标窗口的打开方式
_parent	在上一级窗口打开，常在分帧的框架页面中使用
blank	新建一个窗口打开
self	在同一窗口打开，与默认设置相同
_top	在浏览器的整个窗口打开，将会忽略所有的框架结构

在表 5-1 中提到的框架是一种页面结构，在后面的章节中还会详细讲解。

实例代码：

```
<html>
<head>
<title>目标窗口的打开方式</title>
</head>
<body>
    <center><h2>世界影片介绍</h2></center>
    <hr size=2 color="#990000">
    <font size=4>《孤岛》The Island</font><br><br>
    导演：迈克尔・贝  Michael Bay<br>
    编剧：艾里克斯・库兹曼  Alex Kurtzman<br>
    主演：伊万・迈克格雷戈  Ewan McGregor<br>
          斯嘉丽・约翰森  Scarlett Johansson<br>
          迪蒙・郝德森  Djimon Hounsou<br>
          迈克尔・克拉克・邓肯 Michael Clarke Duncan<br>
```

```
        类型：动作/剧情/科幻/惊悚<br>
        级别：PG-13<br>
        发行：梦工场 Dreamwork<br>
        上映日期：2005 年 7 月 22 日<br>
        官方网站：theisland-themovie.com/<br><br>
        <A href="more.html" target="_blank">影片详情</A>
</body>
</html>
```

设置这段代码的文件名为 index.html，在这段代码中包含一个超链接文本“影片详情”，单击该文字可以打开 more.html 文件。more.html 文件的代码如下：

```
<html>
<head>
<title>目标窗口的打开方式</title>
</head>
<body>
        <center><h2>影片详情</h2></center>
        <font size=4>《孤岛》The Island</font><br><br>
            男青年林肯·6E（伊万·迈克格雷戈）生活在一个貌似乌托邦、却被 21 世纪中叶先进的科学技术设备武装起来的社区里。他跟所有其他社区居民一样，梦想被选中成为“神秘岛”的访客，因为据说那个岛是这个星球上惟一没有被污染过的一片净土。<br>
            但是，不久后的一个意外让林肯惊觉：原来，他头脑中一切关于自己身世和整个这个社区的认识都是一个巨大的谎言！他和社区里的所有居民其实都是“神秘岛”居民们的克隆人，他们的存在只是为了给他们的“原型”提供各种更换用的身体零件，而这个社区其实只是个软禁他们的牢笼……<br>
            得知事情真相之后，林肯意识到：自己迟早会被“收割”，送到“神秘岛”去给主人提供零件。与其像个牲口一样任人宰割，不如拼着命杀出一条血路来！于是乎，林肯和社区另一位居民——美丽的女克隆人乔丹·2D（斯嘉丽·约翰森）一起，踏上了逃离克隆社区、寻找新生活的冒险之旅……
</body>
</html>
```

运行文件 index.html，单击窗口中的“影片详情”链接，可以在一个新的窗口打开文件 more.ht，如图 5-3 所示。

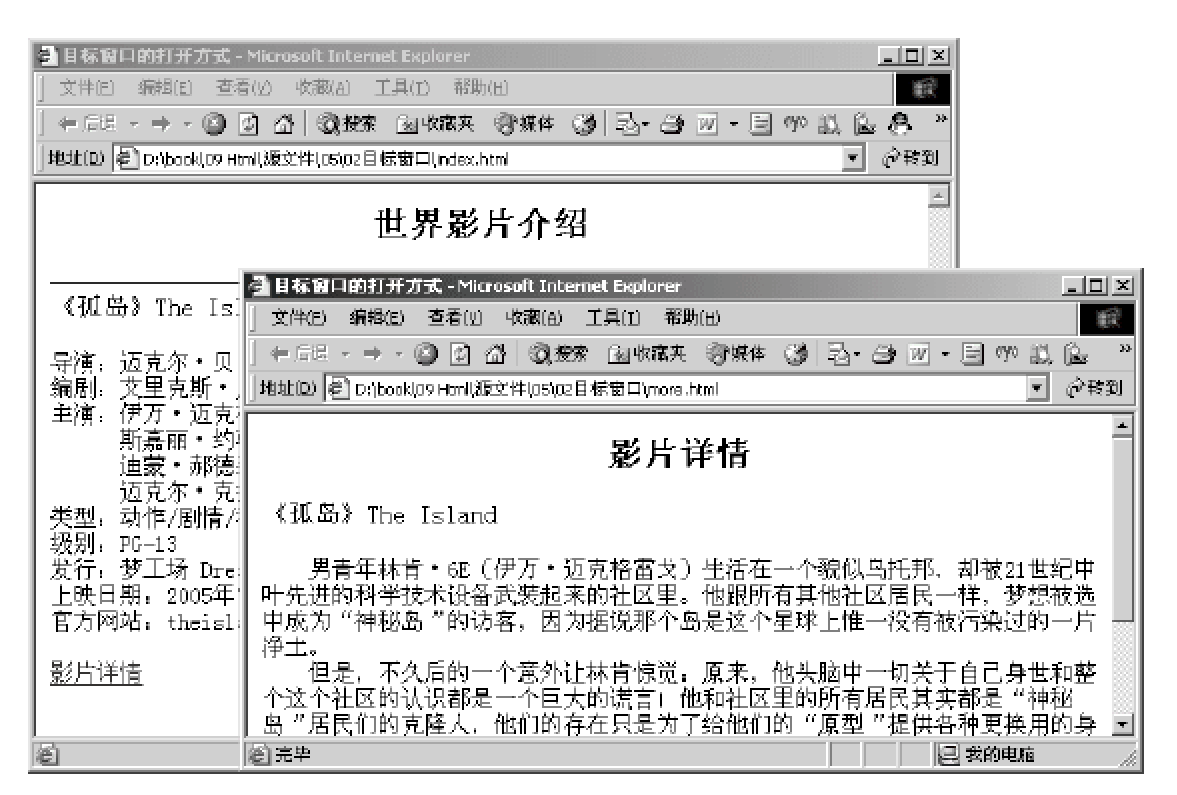


图 5-3　设置目标窗口的打开方式

5.3　内 部 链 接

在各种超链接中，根据链接对象的不同，设置的方式也有所区别。内部链接是指链接的对象是在同一个网站中的资源。下面通过一个实例说明在一个网站中内部链接的实现方法。

在本实例中共包含 3 个文件，分别为 index.html、film1.html 和 film2.html。其中，index.html 作为一个起始页面，另外两个文件放置在与 index.html 文件同级的 film 文件夹中。本实例通过 3 个文件的互相连接说明在网站内部进行链接的方法。

文件 index.html 的代码如下：

```
<html>
<head>
<title>内部链接的实现</title>
</head>
<body>
	<center><h2>世界影片介绍</h2></center>
	<hr size=2 color="#990000">
	<font size=4>征婚启事 Must Love Dogs</font><br><br>
	导演编剧：格雷·大卫·古德伯格 Gary David Goldberg<br>
	主演：戴安·莲恩 Diane Lane<br>
	      克里斯托佛·布拉默 Christopher Plummer<br>
	      约翰·库萨克 John Cusack<br>
	      伊丽莎白·博金斯 Elizabeth Perkins<br>
	类型：爱情<br>
	地区：美国<br>
	上映日期：2005 年 7 月<br><br>
	<A href="film/film1.html">影片详情</A>
	<hr size=3 color="#990066">
	<font size=4>帝企鹅日记 LA MARCHE DE L'EMPEREUR</font><br><br>
	导演/编剧：吕克·贾盖 Luc Jacquet<br>
	主演：Romaine Bohringer<br>
	      Charles Berling<br>
	      Jules Sitruk<br>
	类型：剧情<br>
	地区：法国<br>
	上映日期：2005 年 8 月<br><br>
	<A href="film/film2.html">影片详情</A>
</body>
</html>
```

在该文件中包含了两个链接，一个是影片“征婚启事”的详情介绍的链接，另一个是影片“帝企鹅日记”的介绍。这两个文件都位于文件夹 film 中，在链接时需要在链接地址中加入目录和文件名称。

影片“征婚启事”的介绍文件 film1.html 的代码如下：

```
<html>
<head>
```

```
<title>内部链接的实现</title>
</head>
<body>
<h3>征婚启事 Must Love Dogs</h3>
剧情介绍：<br><br>
    搞对象从来就不是什么轻松的事情。有好事者曾经总结过如下的爱情黄字公式：一份羞辱+两份恼怒+一些对未来"好运气"的盲从=所谓爱情。而现如今的搞对象方式就更没法让人乐观了，网络交友、吃饭相亲、朋友或亲戚的胡乱介绍……到底怎样才能找到合适的意中人啊？<br>
    单身女人莎拉·诺兰(戴安·莲恩)最近正在为搞对象的事情发愁呢。三十锒铛岁的她以前是一名中学女教师，8 个月前刚刚离婚，目前一个人过着孤苦的日子。可想而知，这样的女人要么整天伤心得恨不得患上孤僻症或抑郁症；要么就是脾气暴躁得让所有亲戚朋友敬而远之。无论如何，莎拉的家人是绝不能忍心看她离婚这么长时间还没再找对象的，于是她的两位姐妹出于好心，开始给她介绍各种各样离谱儿的男人来相亲。与此同时，莎拉鳏居的父亲比尔(克里斯托佛·布拉默)也在寻找自己的第二春，最近他看上了一位名叫多丽的思想开放的女人……<br><br>
<A href="../index.html">返回</A>        
<A href="film2.html">下一个影片介绍</A>
</body>
</html>
```

初始页面 index.html 则位于该文件的上一级目录中，在链接时需要在文件名前添加“../”。而影片“帝企鹅日记”的影片介绍与文件 film1.html 位于同一目录下，是同级文件，在链接时只需要将链接地址设置为该文件名 film1.html 即可。文件 film2.html 的代码如下：

```
<html>
<head>
<title>内部链接的实现</title>
</head>
<body>
<h3>帝企鹅日记 LA MARCHE DE L'EMPEREUR</h3>
剧情介绍：<br><br>
    本片记录的是在南极大陆上生活的帝企鹅的生存和繁衍的故事。影片展示了帝企鹅这个滑稽、可爱而又坚强的物种怎样与严酷的自然环境和它们的天敌作斗争，怎样对小企鹅倾注它们的爱，从而完成它们的生命延续之旅。<br>
    在南极洲地冻天寒的冬天，企鹅对生活环境的选择十分有限：一边是北部，这里有广阔的海洋，相对温暖的生存环境，充足的食物来源；一边是南部，寒冷终年、冰雪覆盖，但环境相对稳定。为了使刚出生的小企鹅不受气候突变所带来的危险，帝企鹅只有离开安逸的北部踏上长达几百公里、危险的旅程前往寒冷的南部，在那里产卵，孵卵，直到春回大地。这一切只为了物种的繁衍。这种看起来柔弱笨拙、容易受伤的动物，在从北部前往南部繁殖的九个月的旅程中表现出了顽强的生命力……<br><br>
<A href="../index.html">返回</A>        
<A href="film1.html">下一个影片介绍</A>
</body>
</html>
```

运行初始文件 index.html，会看到页面中包含两个链接文字，如图 5-4 所示。

单击其中一个影片的链接“影片详情”可以打开下一级内容。在这里单击影片“征婚启事”的链接，可以打开如图 5-5 所示的影片介绍窗口。

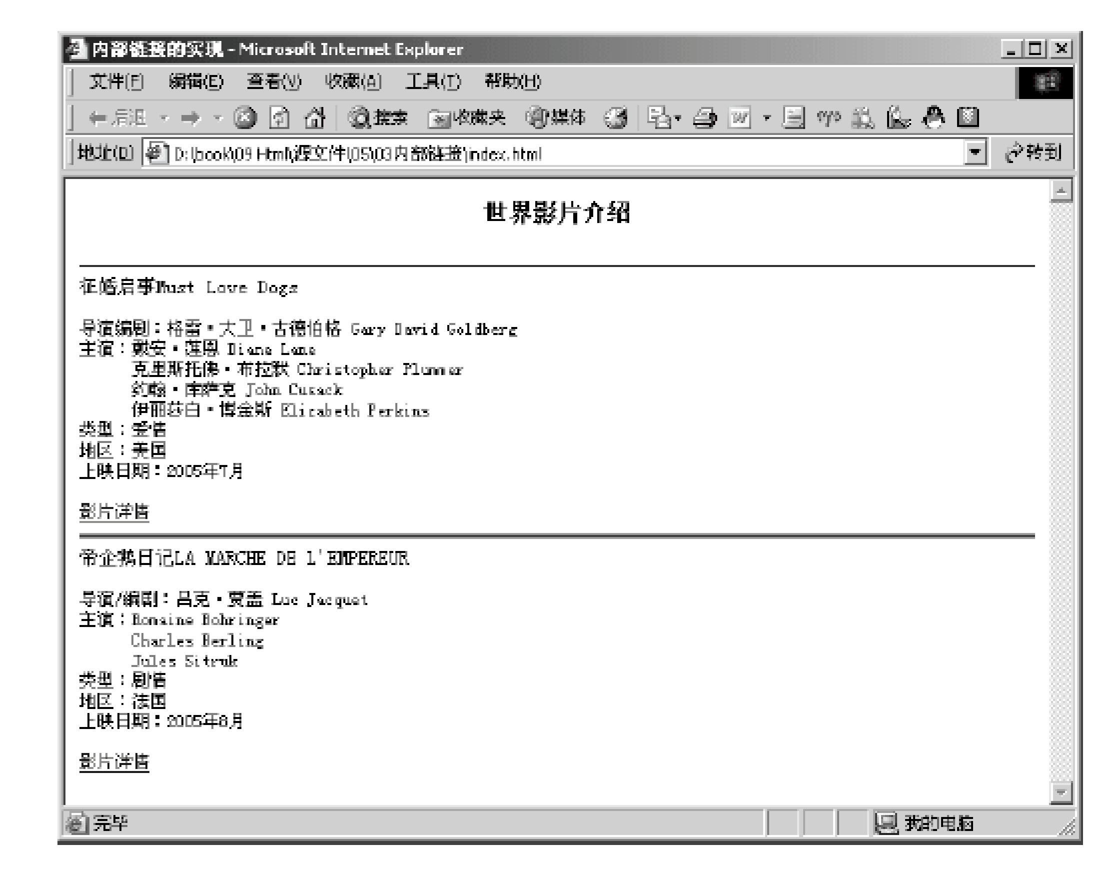


图 5-4　初始窗口的链接效果

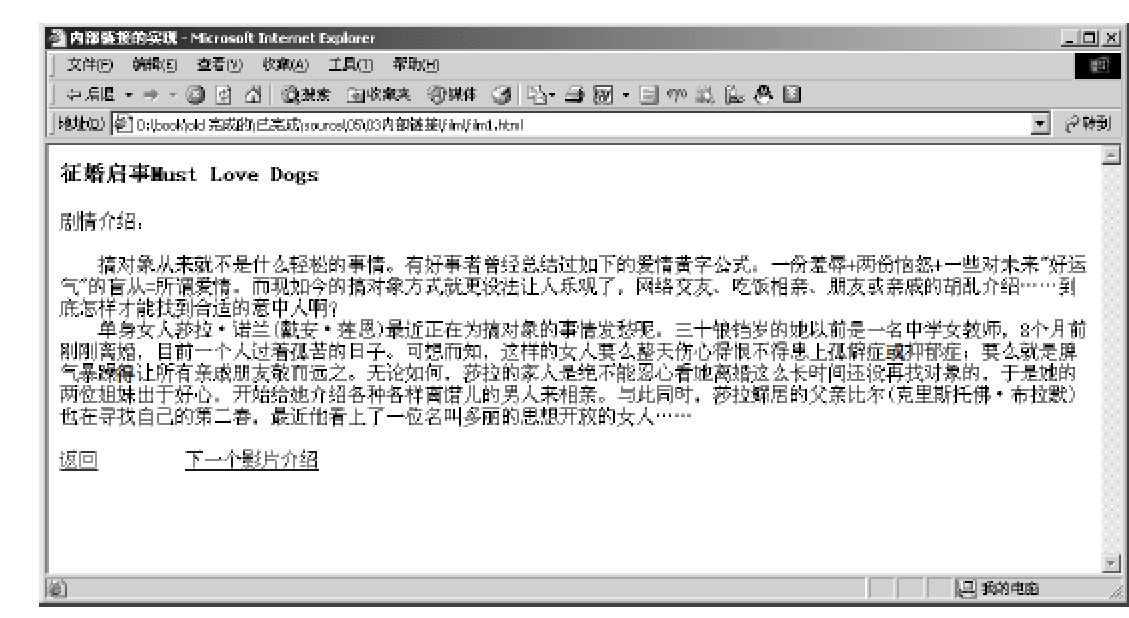


图 5-5　影片介绍窗口

在该窗口中包含了两个链接，单击文本“返回”可以返回到初始页面 index.html 中；单击文本“下一个影片介绍”可以直接打开另外一个影片“帝企鹅日记”的介绍窗口，如图 5-6 所示。

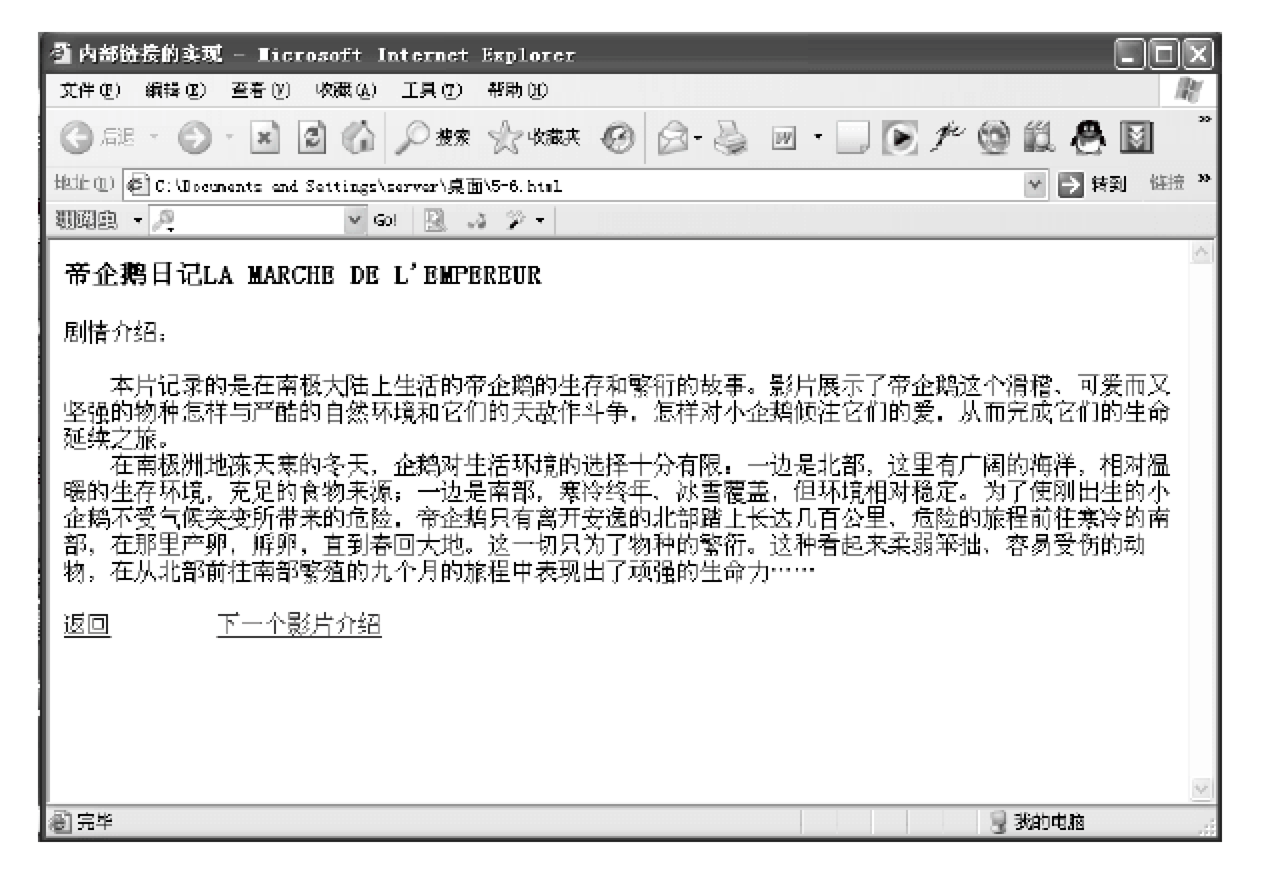


图 5-6　影片“帝企鹅日记”的介绍窗口

5.4　书 签 链 接

当创建的网页内容特别多时，可以通过书签对内容进行链接，这样读者在阅读时可以通过书签进行内容的跳转，这种链接也称为网页内部的书签链接。

5.4.1　建立书签

书签可以与链接文字在同一页面，也可以在不同的页面。但要实现网页内部的书签链接，都需要先建立书签。通过建立的书签才能对页面的内容进行引导和跳转。

语法：<A name="书签名称">文字</A>

说明：在该语法中，书签名称就是对于后面要跳转所创建的书签，而文字则是设置链接后跳转的位置。

实例代码：

```
<html>
<head>
<title>定义书签</title>
</head>
<body>
<h3>爱情测试：你的恋爱狂热度是多少</h3>
<hr size=3>
<A name="answerA">选择 A：凌晨零时到清晨六点</A><br>
     向来朝气蓬勃的你，的确喜欢谈恋爱，喜欢那种活力十足的感觉，但是你的恋爱狂热度，却是平平而已。因为你对生活中的其他事物，如你个人的嗜好，也有着恋爱般的热度，过度分散目标的结果，使你的恋爱狂热度只有 30%<br><br>
<hr size=2>
<A name="answerB">选择 B：清晨六点到中午十二点</A><br>
       你那死不服输的性格将你在人群中独立出来，恋爱的时候，亦因为不肯对对方坦白而遭抛弃，你唯一的好处是你绝不拖拖拉拉没完没了。但是有时候恋爱也要讲小策略的哦，偶尔“认输”也未尝不是件好事！<br><br>
<hr size=2>
<A name="answerC">选择 C：中午十二点到下午六点</A><br>
       你是个十分理智的 MM，但聪明反被聪明误哦，你对每件事都小心谨慎，在爱情道路上，因怕受伤而不敢用情太深，但这会令你的恋人有种被忽视的感觉，对你的热情亦因此大减，其实有时候在他面前装迷糊装可爱，他会更加喜欢你哦。<br><br>
<hr size=2>
<A name="answerD">选择 D：下午六点到凌晨零点</A><br>
       你有点神经质哦，但是是个很执著的 MM，别人很难适应你的处世作风哦，你的男朋友也一样哦，你一时对他风情万种，一时对他冷若冰霜，令他感到十分迷惑。即使对你一见钟情，他亦会被你给他飘忽不定的感觉而吓得溜之大吉，要改变这样的状况，MM 可要好好控制一下自己的情绪哦！
</body>
</html>
```

在这段代码中，定义了 4 个书签，分别命名为 answerA、answerB、answerC 和 answerD。运行这段代码，效果如图 5-7 所示。

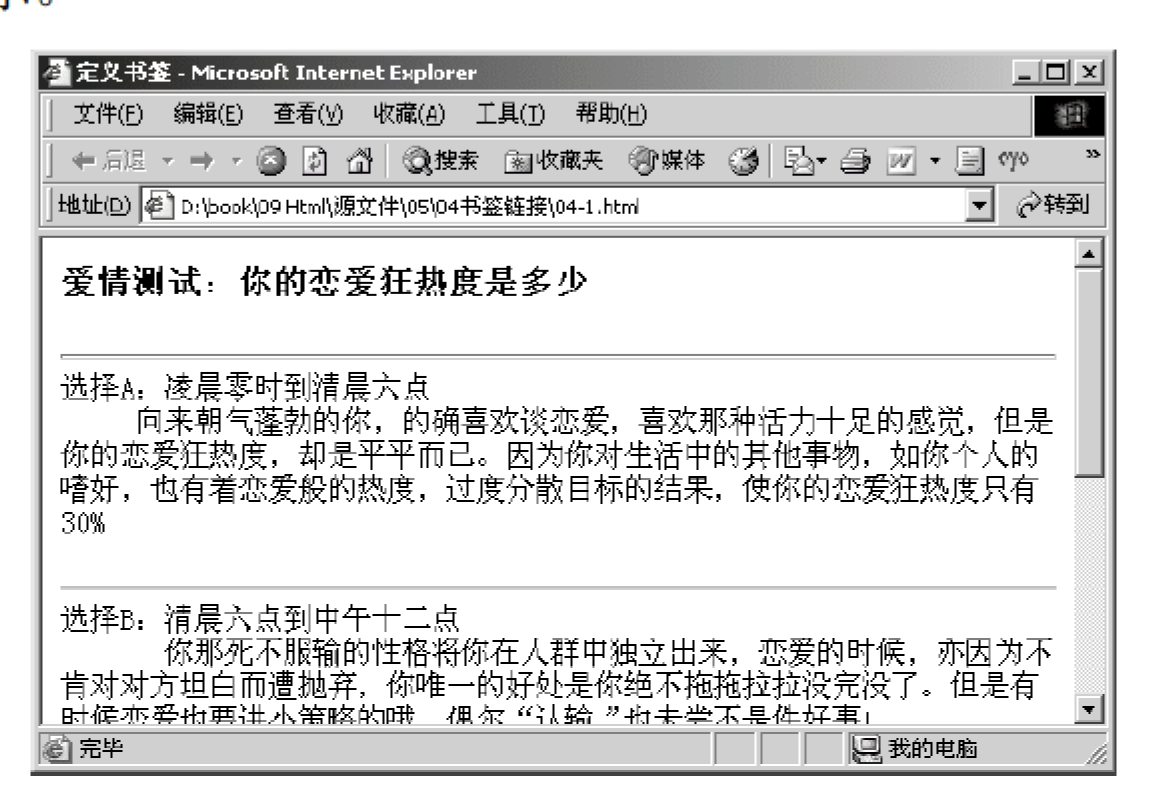


图 5-7　定义书签

可以看到，在浏览器中并不能看到定义的书签，但是它实际上已经存在了。此时就可以定义书签链接了。

5.4.2　链接到同一页面的书签

下面就可以为刚才制作的书签添加链接内容了。在代码的前面增加链接文字和链接地址就能够实现同页面的书签链接。

语法：<A href="#书签的名称">链接的文字</A>

说明：在该语法中，书签的名称就是刚才所定义的书签名，也就是 name 的赋值。而#则代表这是书签的链接地址。

实例代码：

```
<html>
<head>
<title>定义书签</title>
</head>
<body>
<h3>爱情测试：你的恋爱狂热度是多少</h3>
  情窦初开的漂亮小 MM 终于找到了心中的白马王子啦！可是可是……是不是好想好想将感情更顺利地进行下去、急速升温？那么，可爱的 MM 们快来做这个测试吧……保证让你和心中的白马王子开开心心一路牵手到底。一天之中，你最喜欢什么时间哦？<br><br>
<A href="#answerA">A:凌晨零时到清晨六点</A><br><br>
<A href="#answerB">B:清晨六点到中午十二点</A><br><br>
<A href="#answerC">C:中午十二点到下午六点</A><br><br>
<A href="#answerD">下午六点到凌晨零时</A><br><br>
<hr size=3><hr size=3>
<A name="answerA">选择 A：凌晨零时到清晨六点</A><br>
     向来朝气蓬勃的你，的确喜欢谈恋爱，喜欢那种活力十足的感觉，但是你的恋爱狂热度，却是平平而已。因为你对生活中的其他事物，如你个人的嗜好，也有着恋爱般的热度，过度分散目标的结果，使你的恋爱狂热度只有 30%<br><br>
<hr size=2>
<A name="answerB">选择 B：清晨六点到中午十二点</A><br>
       你那死不服输的性格将你在人群中独立出来，恋爱的时候，亦因为不肯对对方坦白而遭抛弃，你唯一的好处是你绝不拖拖拉拉没完没了。但是有时候恋爱也要讲小策略的哦，偶尔“认输”也未尝不是件好事！<br><br>
<hr size=2>
<A name="answerC">选择 C：中午十二点到下午六点</A><br>
       你是个十分理智的 MM，但聪明反被聪明误哦，你对每件事都小心谨慎，在爱情道路上，因怕受伤而不敢用情太深，但这会令你的恋人有种被忽视的感觉，对你的热情亦因此大减，其实有时候在他面前装迷糊装可爱，他会更加喜欢你哦。<br><br>
<hr size=2>
<A name="answerD">选择 D：下午六点到凌晨零点</A><br>
       你有点神经质哦，但是是个很执著的 MM，别人很难适应你的处世作风哦，你的男朋友也一样哦，你一时对他风情万种，一时对他冷若冰霜，令他感到十分迷惑。即使对你一见钟情，他亦会被你给他飘忽不定的感觉而吓得溜之大吉，要改变这样的状况，MM 可要好好控制一下自己的情绪哦！
</body>
</html>
```

运行这段代码，可以看到 4 个文字链接，如图 5-8 所示。

在页面中单击其中的一个链接文字，页面将会跳转到该链接的书签所在位置。单击“A：清晨零时到清晨六点”，跳转后的页面效果如图 5-9 所示。

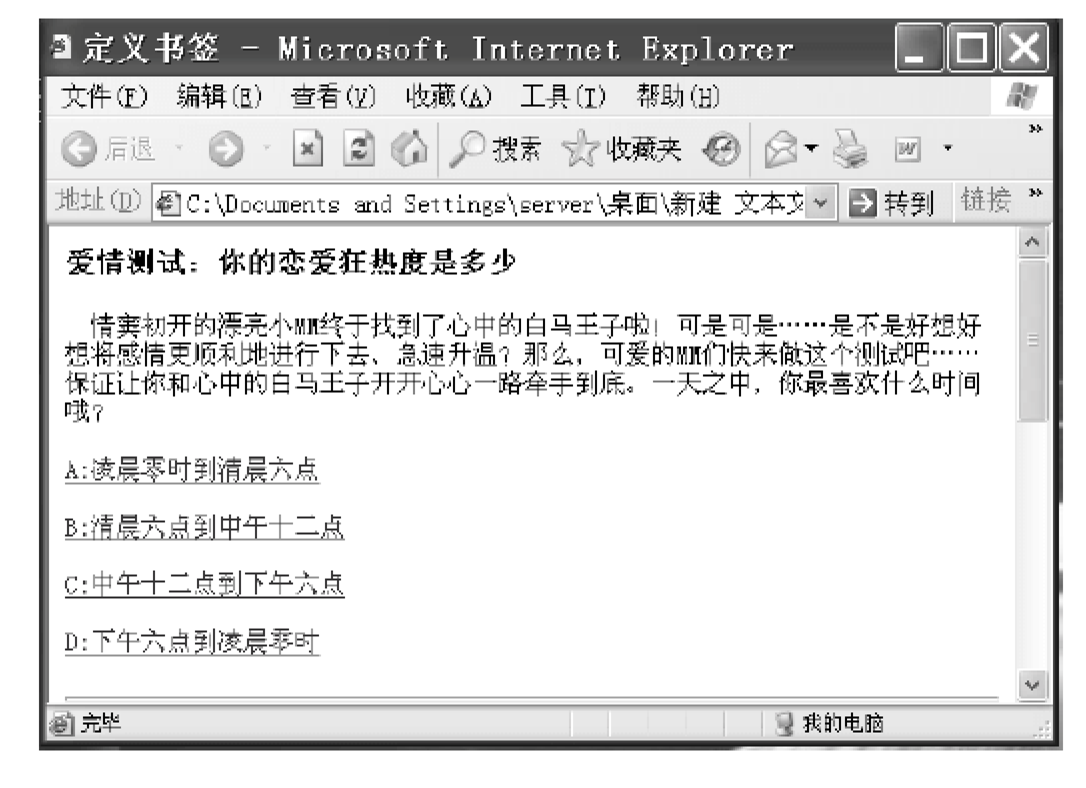


图 5-8　建立书签和链接的页面效果

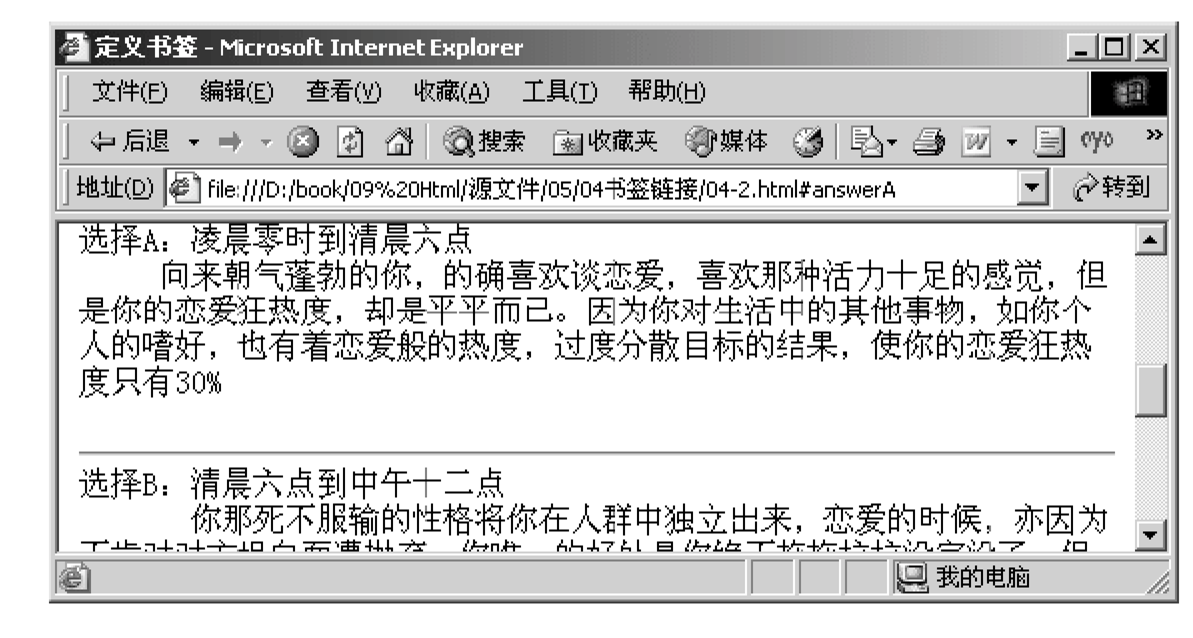


图 5-9　跳转的效果

单击页面中的“C：中午十二点到下午六点”，页面跳转到书签 answerC 所在的位置，如图 5-10 所示。

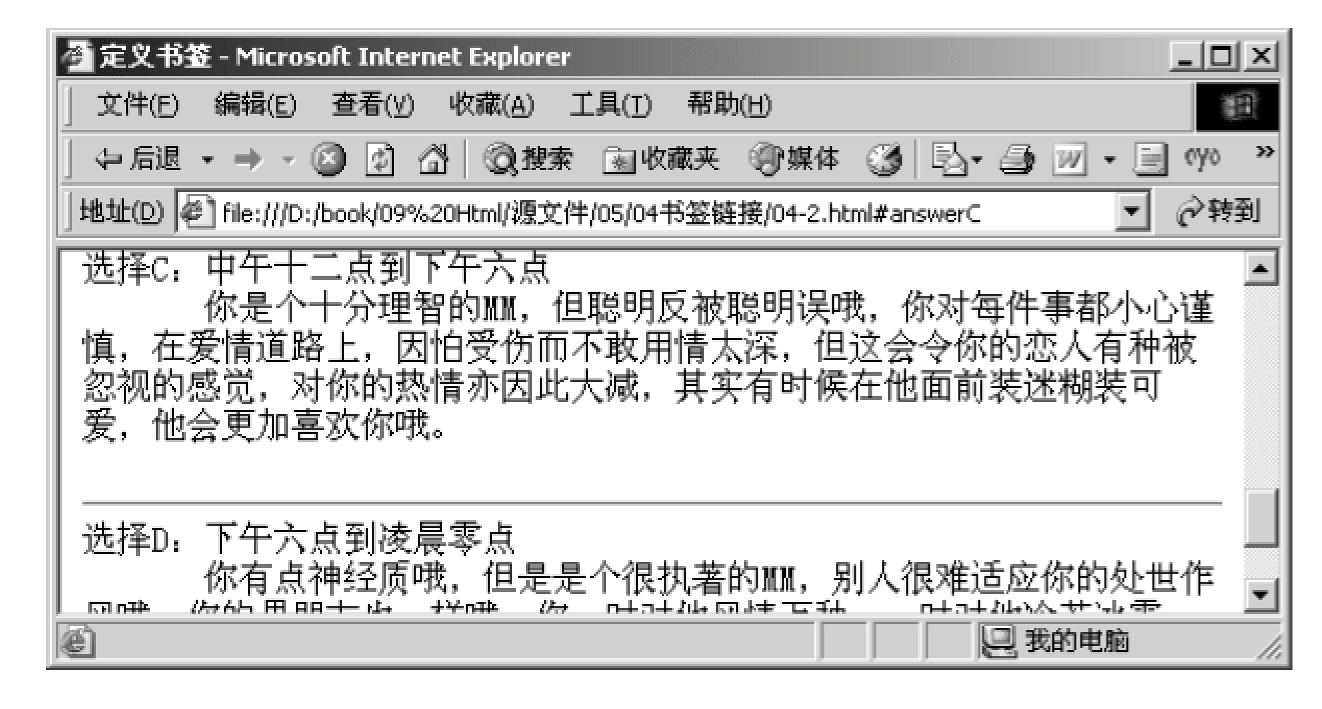


图 5-10　跳转到书签所在位置

5.4.3　链接到不同页面的书签

书签链接不但可以链接到同一页面，也可以在不同页面中设置。

语法：<A href="链接的文件地址#书签的名称">链接的文字</A>

说明：在该语法中，与同一页面的书签链接不同的是，需要在链接的地址前增加文件所在的位置。

下面设置一个单独的链接页面，使其链接到前面定义的书签页面。

实例代码：

```
<html>
<head>
<title>定义书签</title>
</head>
```

```
<body>
<h3>爱情测试：你的恋爱狂热度是多少</h3>
  情窦初开的漂亮小 MM 终于找到了心中的白马王子啦！可是可是……是不是好想好想将感情更顺利地进行下去、急速升温？那么，可爱的 MM 们快来做这个测试吧……保证让你和心中的白马王子开开心心一路牵手到底。一天之中，你最喜欢什么时间哦？<br><br>
<A href="04-1.html#answerA">A：凌晨零时到清晨六点</A>
        
<A href="04-1.html#answerB">B：清晨六点到中午十二点</A><br><br>
<A href="04-1.html#answerC">C：中午十二点到下午六点</A>
        
<A href="04-1.html#answerD">D：下午六点到凌晨零时</A><br><br>
</body>
</html>
```

运行这段代码，效果如图 5-11 所示。

单击其中的某个链接，例如“B：清晨六点到中午十二点”，就可以直接链接到书签所在的位置，如图 5-12 所示。

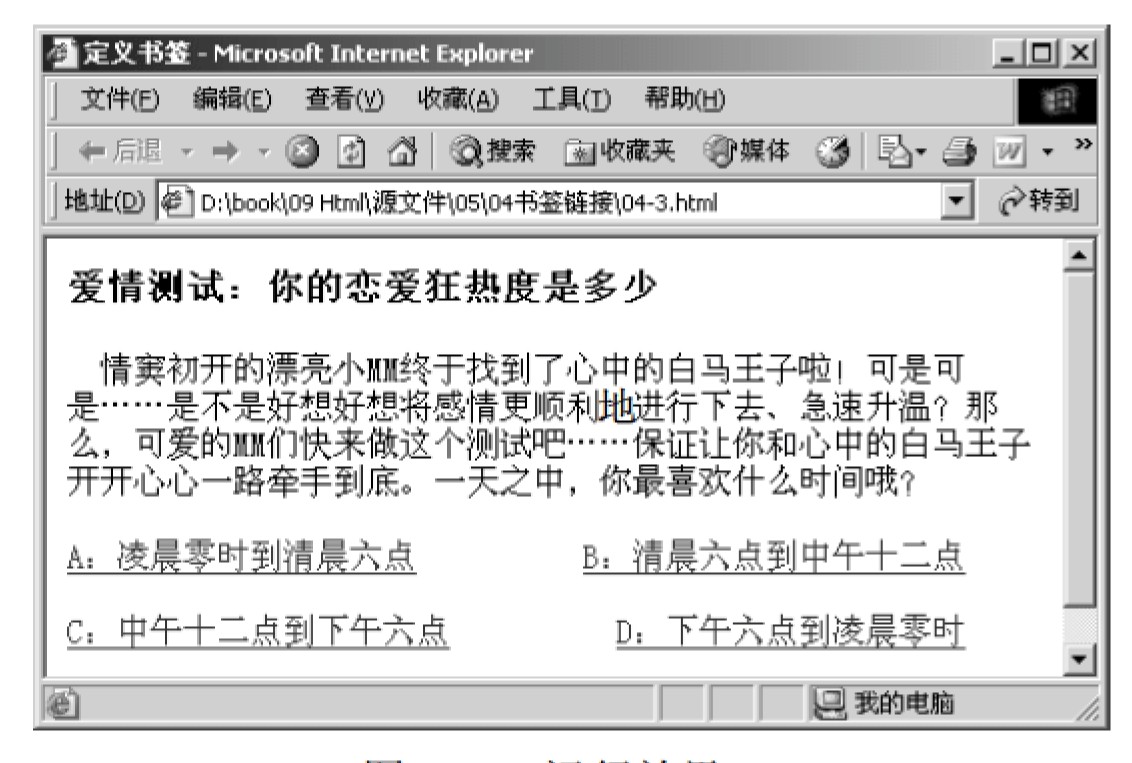


图 5-11　运行效果

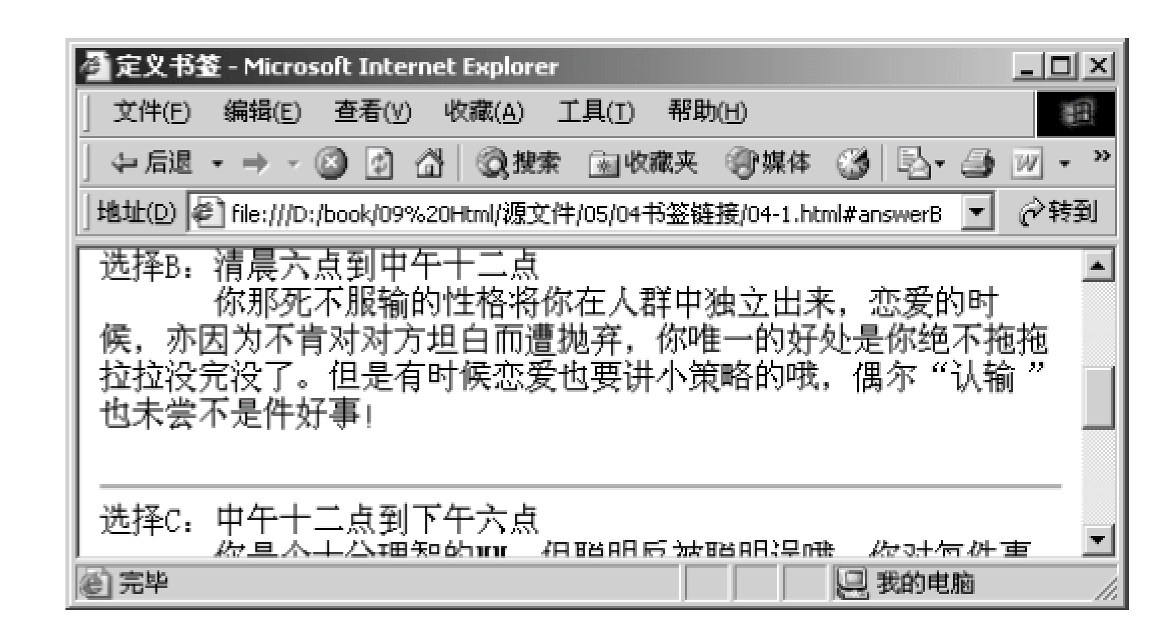


图 5-12　链接的效果

5.5　外 部 链 接

外部链接是指跳转到当前网页之外的资源中，例如其他网站中某个页面或元素。这种链接在设置时一般需要书写绝对地址，最常见的是使用 URL 统一资源定位符“http://”来表示。此外还有一些其他的格式，见表 5-2。

表5-2　绝对地址的设置格式

格　　式	表示的含义
http://	采用WWW服务进入万维网站点
ftp://	通过FTP访问文件传输服务器
telnet://	启动Telnet
mailto://	直接启动邮件系统Email

5.5.1　通过 HTTP 协议

网页中最常使用 HTTP 协议进行外部链接的是在设置友情链接时。

语法：<A href="http://……">链接文字</A>

说明：在该语法中，http://表明这是关于 HTTP 协议的外部链接，而在其后则输入网站的网址即可。

实例代码：

```
<html>
<head>
<title>唐诗宋词欣赏</title>
</head>
<body>
<font size=5 color="#990066">浣溪沙</font><br><br>
苏轼<br>
游蕲水清泉寺，寺临兰溪，溪水西流。<br>
山下兰芽短浸溪，松间沙路净无泥。<br>
萧萧暮雨子规啼。谁道人生无再少？<br>
门前流水尚能西！休将白发唱黄鸡。<br>
<hr size=2>
<A href=" http://www.shiandci.net ">更多唐诗宋词，请进入……</A>
</body>
</html>
```

运行这段代码，可以实现如图 5-13 所示的页面效果。

在页面中包含一个文本链接，它所链接的地址就是万维网上的一个网址，单击链接文字可以打开如图 5-14 所示的网站。

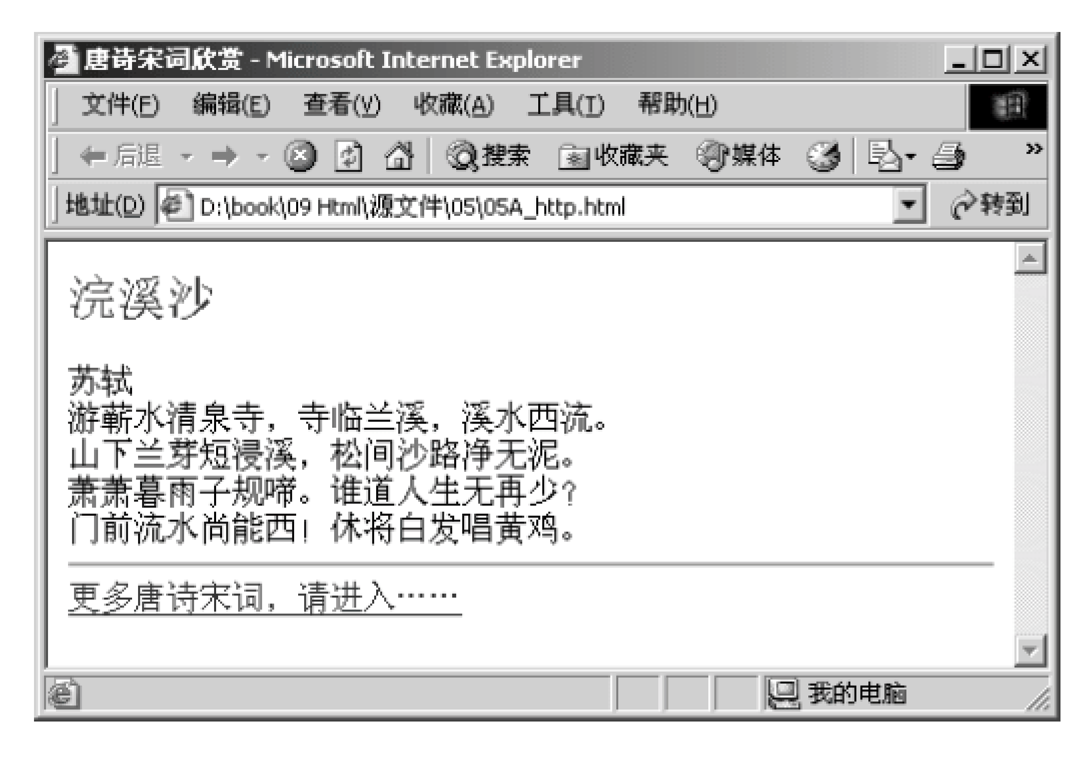


图 5-13　设置外部链接的页面

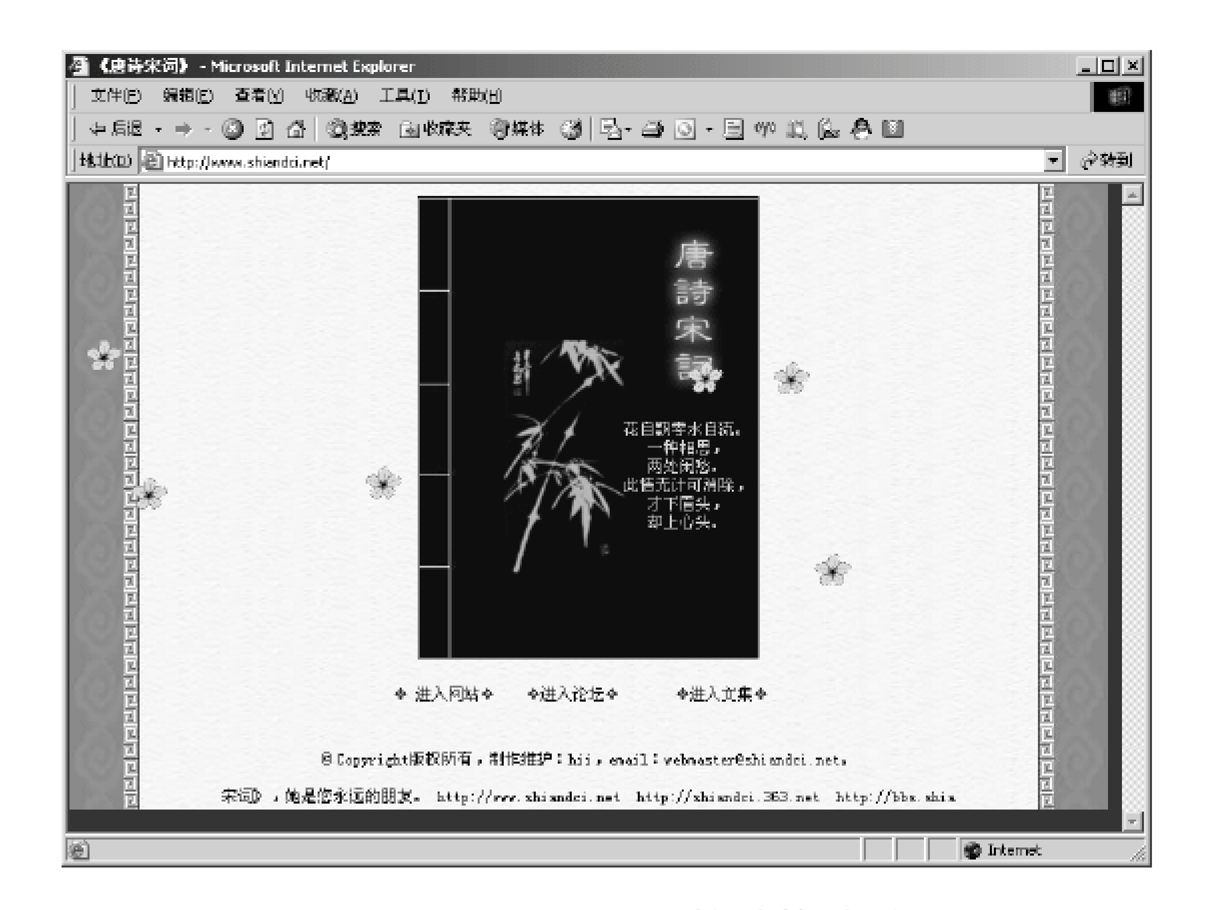


图 5-14　打开的链接地址

5.5.2　通过 FTP 协议

网络中还存在一种 FTP 协议，这是一种文件传输协议。在很多 FTP 地址中，可以获得许多有用的

软件和共享文件。FTP 需要获得许可才能在网络上传播，因此需要从服务器管理员处取得登录的权限。有一些 FTP 服务器可以匿名访问，同样能够获得一些公开的数据。

语法：<A href="ftp://……">链接文字</A>

说明：在该语法中，ftp://表明这是关于 FTP 协议的外部链接，而在其后则输入网站的网址即可。

实例代码：

```
<html>
<head>
<title>使用 FTP 服务</title>
</head>
<body>
进行 FTP 服务器的链接：
<hr size=2>
<A href="ftp://ftp.pku.edu.cn" target="_blank">进入北京大学 FTP 站点</A>
</body>
</html>
```

运行这段代码，可以实现如图 5-15 所示的页面效果。

在页面中包含一个文本链接，它所链接的地址就是一个 FTP 网址。而设置的 target 参数则决定了在打开的新页面中进行链接。打开的效果如图 5-16 所示。

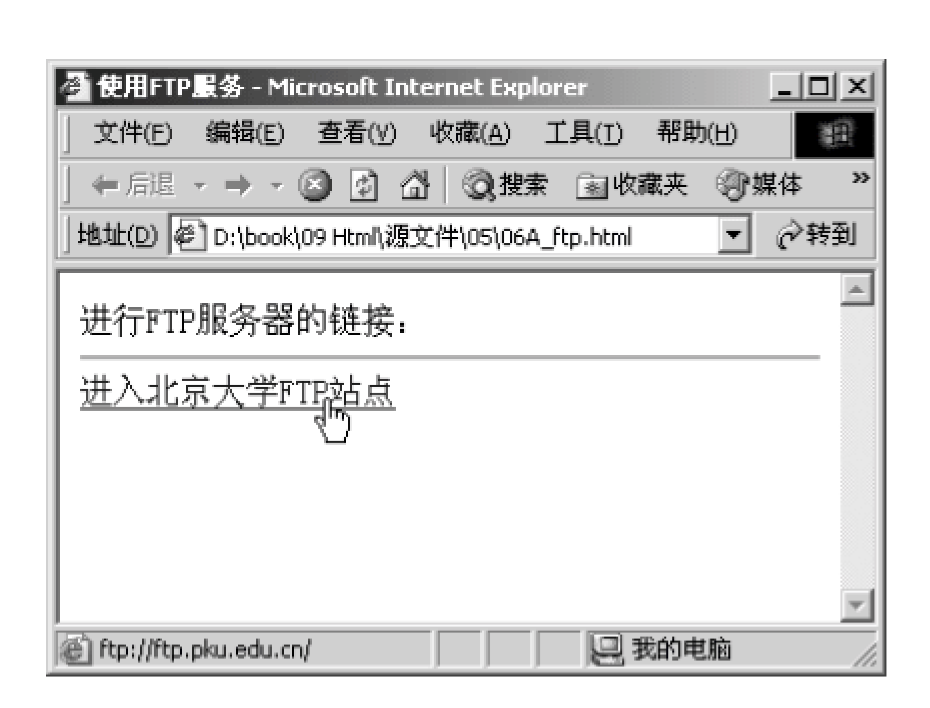


图 5-15　设置 FTP 链接的页面

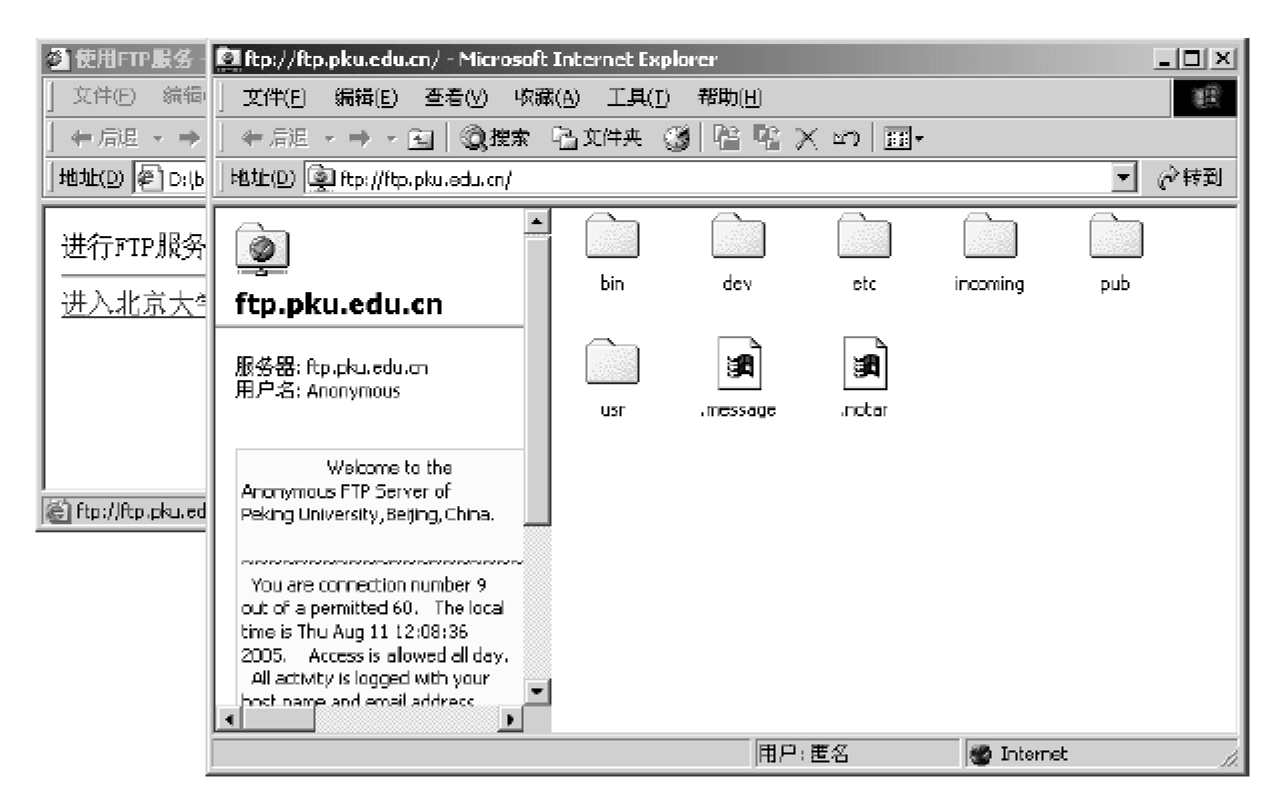


图 5-16　打开的 FTP 地址

5.5.3　通过 Telnet 链接

Telnet 常常用来登录一些 BBS 网址，也是一种远程登录方式。

语法：<A href="telnet://地址">链接文字</A>

说明：这种链接方式与其他两种类似，不同的就是它登录的是 Telnet 站点。

实例代码：

```
<html>
<head>
<title>建立 Telnet 链接</title>
</head>
```

```
<body>
       清华大学的前身是清华学堂，始建于 1911 年，曾是由美国“退还”的部分“庚子赔款”建立的留美预备学校。1912 年更名为清华学校，1925 年设立大学部，开始招收四年制大学生，同年开办研究院（国学门），1928 年更名为“国立清华大学”，并于 1929 年秋开办研究院，各系设研究所。1937 年抗日战争爆发后，南迁长沙，与北京大学、南开大学联合办学，组建国立长沙临时大学，1938 年迁至昆明，改名为国立……<br><br>
<hr>
<A href="telnet://166.111.8.238">清华大学 BBS 站点</A>
</body>
</html>
```

运行这段代码，效果如图 5-17 所示。

单击页面中的链接文字，将会启动 Telnet 工具进入该 BBS 站点，如图 5-18 所示。

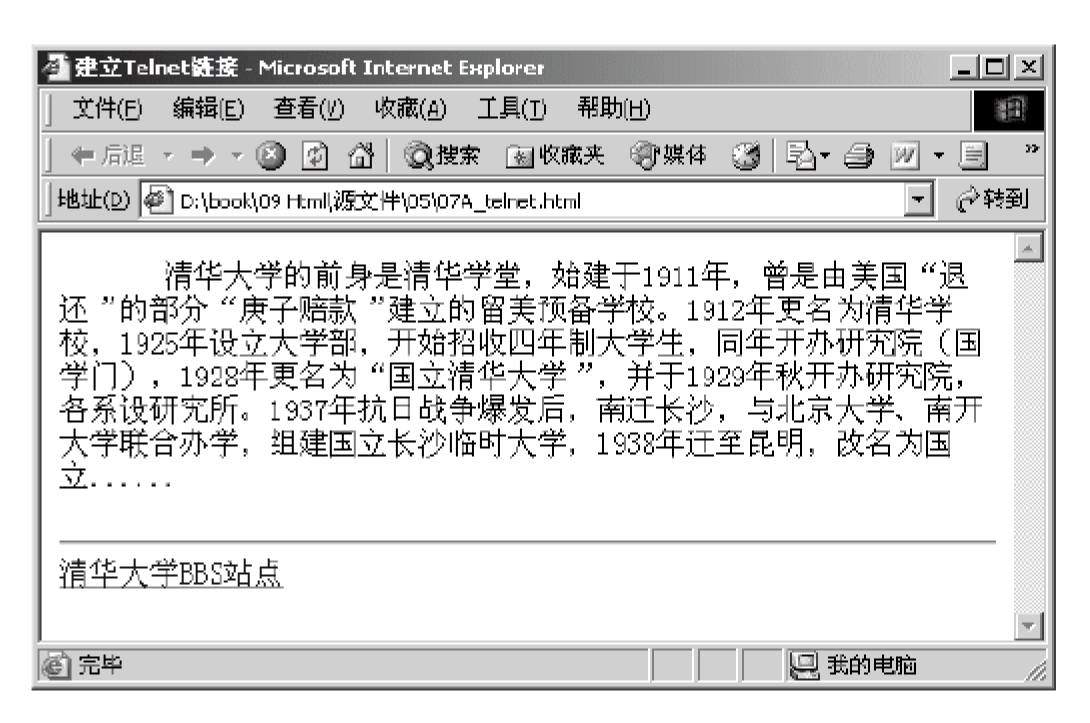


图 5-17　Telnet 链接页面

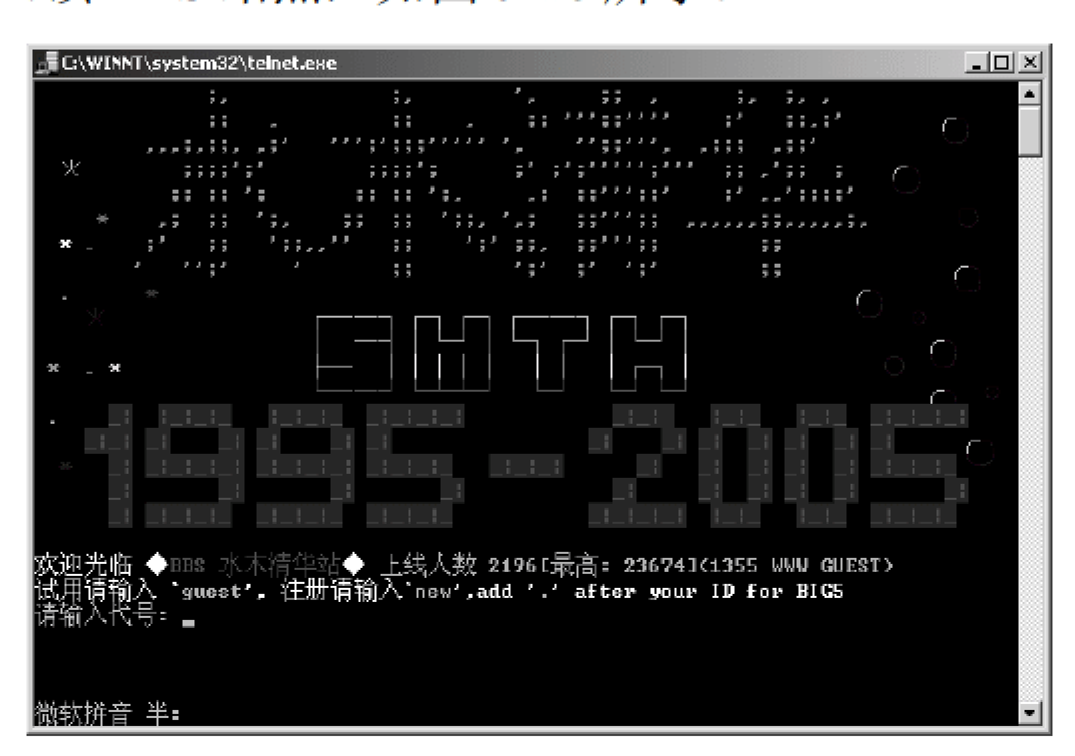


图 5-18　水木清华 BBS 站点

5.5.4　发送到 Email

在网络中，很多拥有个人网页的朋友都喜欢在网站的醒目位置处写上自己的电子邮件地址，这样网页浏览者一旦用鼠标单击一下由 mailto 组成的超链接后，就能自动打开当前计算机系统中默认的电子邮件客户端软件，例如 Outlook Express、Foxmail 等。其实这是通过 mailto 标签来实现的。

语法：<A href="mailto:电子邮件地址">链接文字</A>

说明：在该语法的电子邮件地址后还可以增加一些参数，见表 5-3。

表5-3　mailto标签的参数

参　　数	表示的含义	语　　法
CC	抄送收件人	<A href="mailto:电子邮件地址?CC=电子邮件地址">链接文字</A>
Subject	电子邮件主题	<A href="mailto:电子邮件地址?Subject=主题文字">链接文字</A>
BCC	暗送收件人	<A href="mailto:电子邮件地址?BCC=电子邮件地址">链接文字</A>
Body	电子邮件内容	<A href="mailto:电子邮件地址?Body=邮件内容">链接文字</A>

这些参数可以没有，也可以同时设置几个。在带有多个参数时，需要使用&符号对参数进行分隔。

实例代码：

```
<html>
<head>
<title>发送电子邮件</title>
</head>
<body>
   如果您在使用网站的时候发现了问题或者 Bug，欢迎您给我们提出。<br>
<A href="mailto:bug@sina.com?BCC=xyf@163.com">发现问题</A><br><br>
   如果您对我们的工作有建议或意见，也欢迎您来信提出。<br>
< A href="mailto:opinion@sina.com?CC=xyf@163.com&Subject=意见和建议">提出意见建议</A>
<br><br>
<A href="mailto:xyf@163.com">给作者来信</A>
</body>
</html>
```

运行这段代码，效果如图 5-19 所示。

单击页面中的链接文字“发现问题”，可以打开默认的电子邮件软件 Outlook Express，如图 5-20 所示。在软件中可以看到，除了设置的 Email 地址之外，在“密件抄送”文本框中也设置了电子邮件的地址。

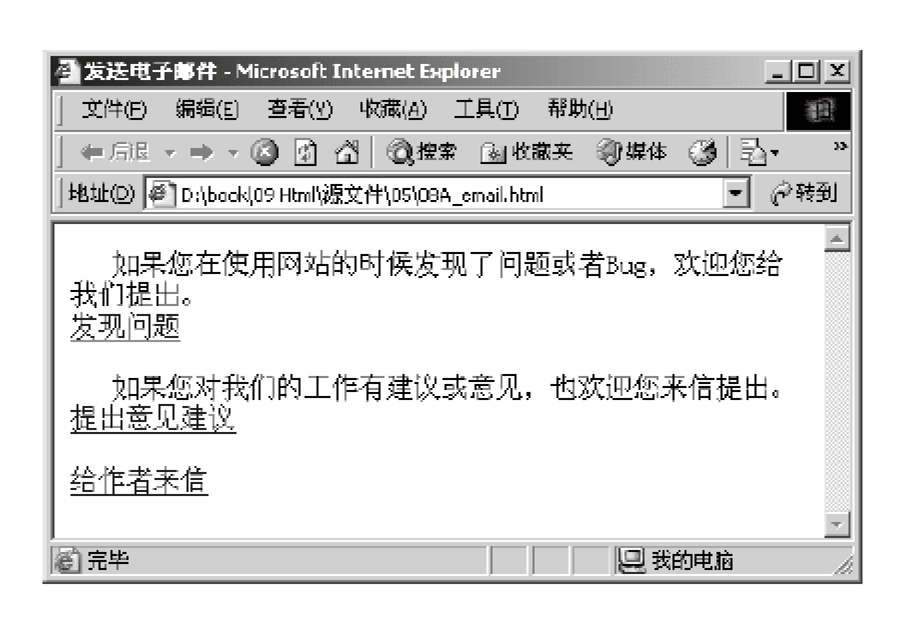


图 5-19　设置 Email 链接的页面



图 5-20　发送电子邮件

单击页面中的“提出意见建议”文字链接，在打开的 Outlook Express 中，除了“收件人”、“抄送”文本框中设置了电子邮件外，还同时设置了邮件的主题，如图 5-21 所示。

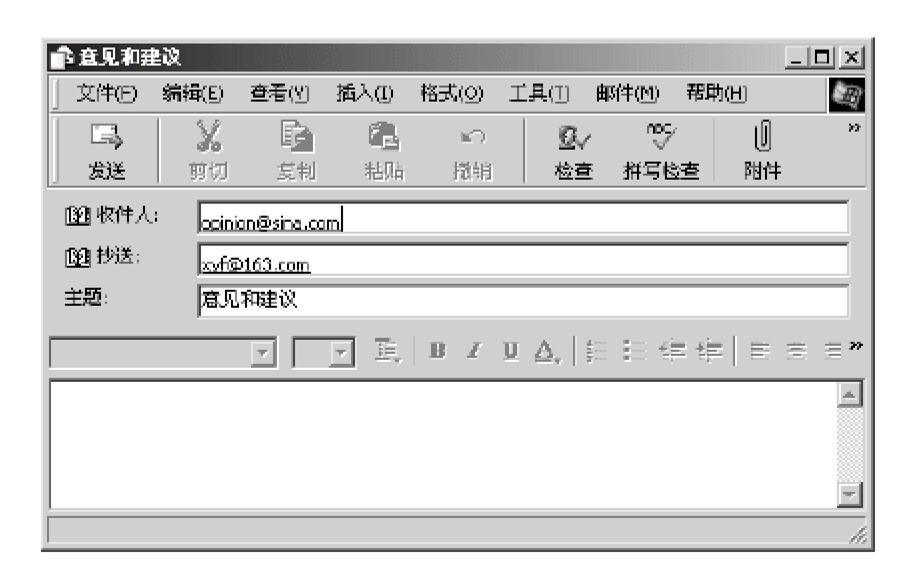


图 5-21　发送邮件设置

5.5.5　下载文件

在浏览网页时下载一些文件也是常有的事情，在某些网站中，只需要单击一个链接就可以自动下

载文件，非常方便，这也可以使用文本链接来实现。

语法：<A href="文件所在地址">链接文字</A>

说明：在文件所在地址中设置文件的路径，可以是相对地址，也可以是绝对地址。

实例代码：

```
<html>
<head>
<title>文件的下载</title>
</head>
<body>
<h4>网际快车 FlashGet</h4>
<A href="file.exe">软件下载试用</A><br><br>
网际快车 FlashGet 通过把一个文件分成几个部分同时下载可以成倍地提高速度，下载速度可以提高 100%到500%。FlashGet 可以创建不限数目的类别，每个类别指定单独的文件目录，不同的类别保存到不同的目录中去，强大的管理功能包括支持拖拽、更名、添加描述、查找、文件名重复时可自动重命名等，而且下载前后均可轻易管理文件。
</body>
</html>
```

运行这段代码，效果如图 5-22 所示。

单击页面中的文本链接“软件下载试用”，可以打开如图 5-23 所示的提示对话框。

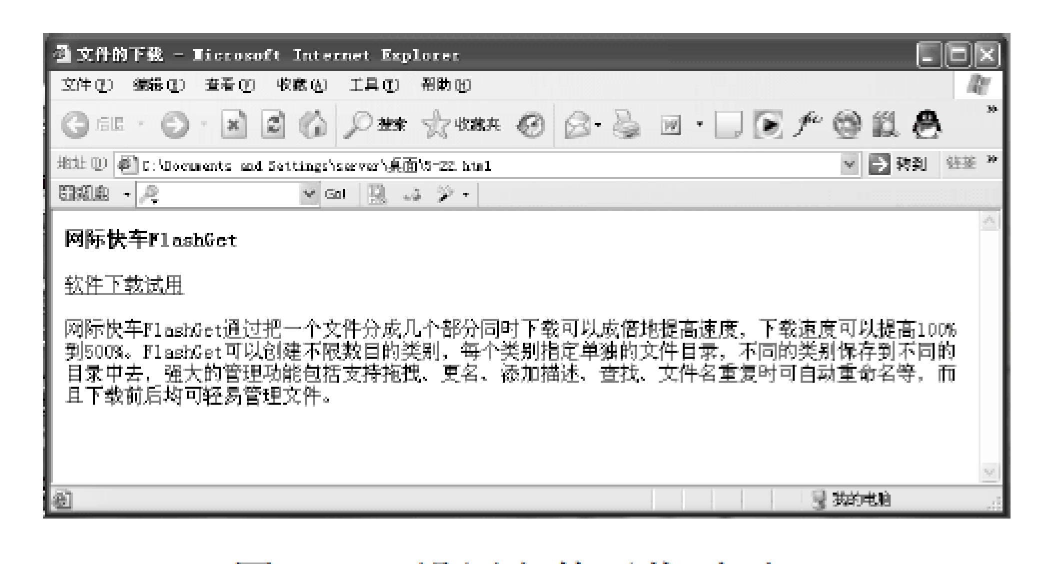


图 5-22　设置文件下载页面

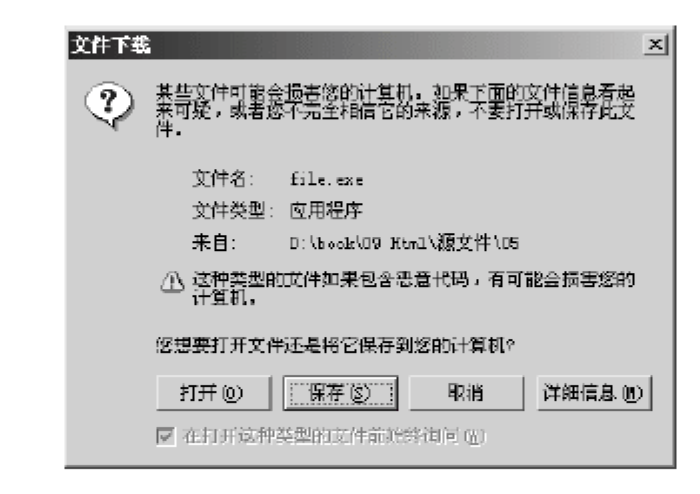


图 5-23　文件下载的提示对话框

在该对话框中可以单击“打开”按钮直接打开软件；也可以单击“保存”按钮将该文件保存到磁盘。如果选择保存文件到磁盘，将会打开如图 5-24 所示的对话框。在对话框中设置相应的存储位置，单击“保存”按钮即可实现文件的保存。



图 5-24　另存为对话框

第 6 章

使用图像

- 图像格式
- 添加图像——img
- 设置图像属性
- 图像的超链接

万维网（World Wide Web）与其他网络类型（如 FTP）最大的不同就在于它在网页上可呈现丰富的色彩及图像。用户可以在网页中放入自己的照片；可以放入公司的商标；还可以把图像作为一个按钮来链接到另一个网页，这就让网页变得丰富多彩了。

6.1　图 像 格 式

每天在网络上交流的计算机多不胜数，因此使用的图像格式一定能够被每一个操作平台所接受。当前万维网上流行的图像格式以 GIF 及 JPEG 为主，另外还有一种名为 PNG 的文件格式，也被越来越多地应用在网络中。以下分别对这 3 种图像格式的特点进行介绍。

1．GIF 格式

GIF 格式采用 LZW 压缩，是以压缩相同颜色的色块来减少图像大小的。由于 LZW 压缩不会造成任何品质上的损失，而且压缩效率高，再加上 GIF 在各种平台上都可使用，所以很适合在互联网上使用，但 GIF 只能处理 256 色。

GIF 格式适合于商标、新闻式的标题或其他小于 256 色的图像。要想将图像以 GIF 的格式存储，可参考下面范例的方法。

LZW 压缩是一种能将数据中重复的字符串加以编码制作成一数据流的一种压缩法，通常应用于 GIF 图像文件的格式。

2．JPEG 格式

对于照片之类全彩的图像，通常都以 JPEG 格式来进行压缩，也可以说，JPEG 格式通常用来保存超过 256 色的图像格式。JPEG 的压缩过程会造成一些图像数据的损失，所造成的“损失”是剔除一些视觉上不容易察觉的部分。如果剔除适当，视觉上不但能够接受，而且图像的压缩效率也会提高，使图像文件变小；反之，剔除太多图像数据，则会造成图像过度失真。

3．PNG 格式

PNG（Portable Network Graphics）图像格式是一种非破坏性的网页图像文件格式，它提供了将图像文件以最小的方式压缩却又不造成图像失真的技术。它不仅具备了 GIF 图像格式的大部分优点，而且还支持 48-bit 的色彩，更快的交错显示，跨平台的图像亮度控制，更多层的透明度设置。

6.2　添加图像——img

有了图像文件之后，就可以使用 img 标记将图像插入到网页中，从而达到美化页面的效果。

语法：<img src="图像文件的地址">

说明：在该语法中，src 参数用来设置图像文件所在的路径，这一路径可以是相对路径，也可以是绝对路径。

实例代码：

```
<html>
<head>
<title>插入图像文件</title>
</head>
<body>
```

```
        <h3>动漫世界，精彩无限！</h3>
        动漫名称：魔卡少女樱<br>
          打开封印之书的木之本樱，因身负回收库洛牌的任务而变身成拥有魔法的魔卡少女，就此展开了一系列的冒险。动画的基本架构与漫画大致相同，在细节部分做了一些改动，并出现了原作中没有的卡片和角色，卡片收集的顺序上也有所变化，剧情节奏较漫画延缓不少。<br>
        <!--在页面中居中插入一张卡通图片-->
        <center>
            <img src="pic.jpg">
        </center>
</body>
</html>
```

运行这段代码，可以看到页面中插入了图片，如图 6-1 所示。如果不通过
标记进行换行操作，那么图片不会自动换行。

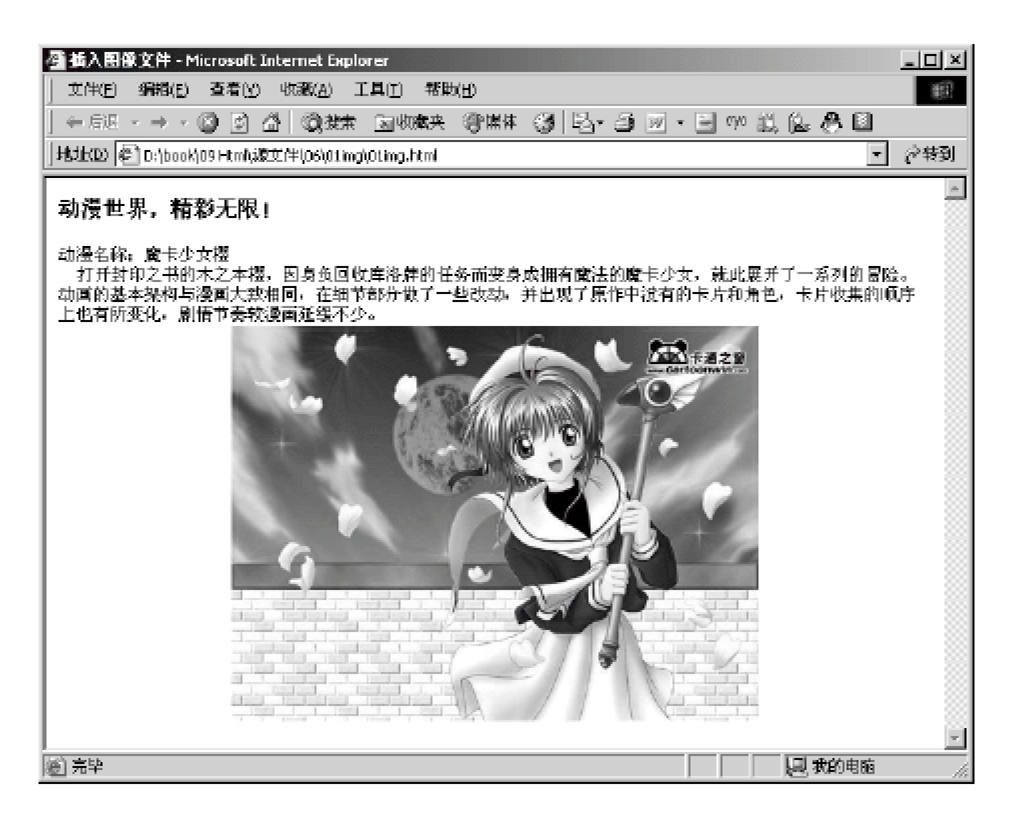


图 6-1　插入图片的效果

6.3　设置图像属性

在网页中直接插入图片时，图像的大小和原图是相同的，而在实际应用时可以通过各种图像属性的设置调整图像的大小、分辨率等内容。

6.3.1　图像高度——height

通过 height 属性可以设置图片显示的高度，默认情况下，改变高度的同时，其宽度也会等比例进行调整。

语法：<img src="图像文件的地址" height=图像的高度>

说明：在该语法中，图像的高度单位是像素。

实例代码：

```
<html>
<head>
```

```
<title>设置页面图像大小</title>
</head>
<body>
    <h3>电视剧：仙剑奇侠传</h3>
       根据 RPG 游戏改编，故事讲述的是平凡少年逍遥与婶婶相依为命，经营着一家小客栈。有一天，婶婶突然生了重病，焦急的逍遥前往仙灵岛寻找解药。仗着从剑仙学得的几招剑法，逍遥踏上了一段危险重重的旅程，开始了一段波澜壮阔的冒险与奇遇……<br>
    <!--在页面中居中插入两张图片-->
    <center>
        <!--默认的图片大小-->
        <img src="pic.jpg">
        <!--设置图片的高度为 160 像素-->
        <img src="pic.jpg" height=160>
    </center>
</body>
</html>
```

运行这段代码，可以看到设置了高度的图片大小改变的效果，如图 6-2 所示。

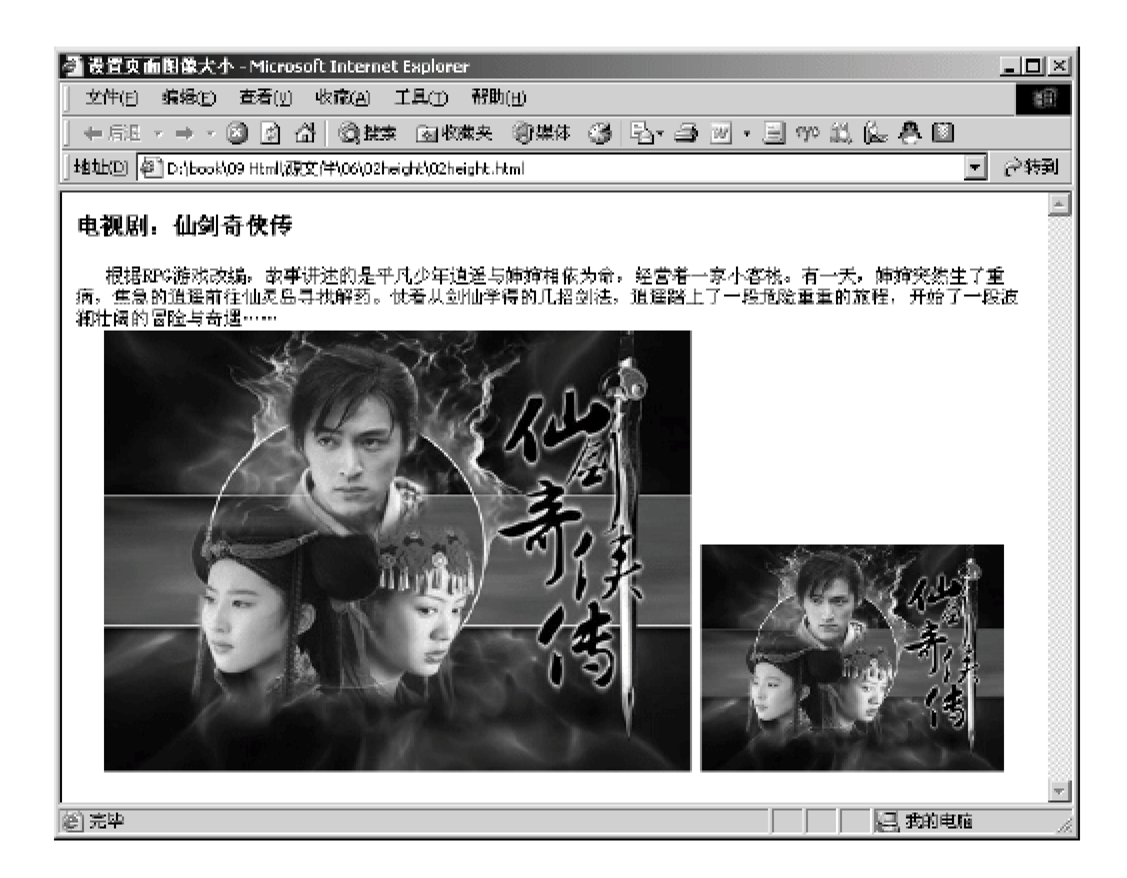


图 6-2　调整图片的高度

6.3.2　图像宽度——width

图像宽度的属性与图像高度类似，同样是用来调整图像大小的。

语法：<img src="图像文件的地址" width=图像的宽度>

说明：在该语法中，图像的宽度单位是像素。如果在使用属性的过程中，只设置了高度或宽度，则另外一个参数会等比例变化。如果同时设置两个属性，且缩放比例不同的情况下，图像很可能会变形，下面通过实例说明。

实例代码：

```
<html>
<head>
<title>设置页面图像大小</title>
</head>
<body>
    <h3>轩辕剑三外传：天之痕</h3>
    <hr size=2>
        神州大地上，从神话时代流传下来十种上古神器——钟、剑、斧、壶、塔、琴、鼎、印、镜、石。它们各自有着迥然不同的绝世力量。只要稍加利用即可纵横四海，无敌天下。但它们的下落，已湮灭于神州漫长之乱世历史中。<br>
       除了轩辕剑，还有创世神开天辟地使用的神器炼妖壶，在上古英雄的手中辗转流传，在这些古人的庇佑下，中国到了文化鼎盛的时代——隋唐。<br>
       公元七世纪，南北朝时期，北朝皇帝隋文帝派兵消灭了南朝陈国后，结束了中原南北朝数百年之分裂局面，重新统一了中国并成立了隋国。然而陈国的遗民悲痛祖国亡灭，在江南集结大军数万兴兵反抗，企图一举复国。隋文帝下令出兵平乱，然而最令人惊讶的是，这支平乱部队为首竟是一位身披神秘斗篷，年近十二岁的少年。他手持一把雕有古朴花纹的金色之剑，此乃十大上古神器之一——轩辕剑。少年在一击之间，便
```

```
将反抗朝廷的陈国数万人马化为飞灰！自此，大隋威名让所有反抗者闻之色变，再没有人敢轻起反抗之念。<br>
        天之痕的故事从十六年后开始，发生在轩辕剑三的一百三十三年前……
<br>
     <!--在页面中居中插入两张图片-->
     <center>
          <!--设置图片的宽度为 160 像素-->
          <img src="pic.jpg" width=260>
          <!--同时设置图片的高度和宽度-->
          <img src="pic.jpg" width=260 height=140>
     </center>
</body>
</html>
```

运行这段代码，可以看到设置了宽度的图片大小改变的效果，如图 6-3 所示。在该图中，第一张图片只设置了图像的宽度，因此它被等比例缩小。而在第二张图片中，同时设置了图片的高度和宽度，而且缩小的比例不同，因此造成了图片被压扁的效果。

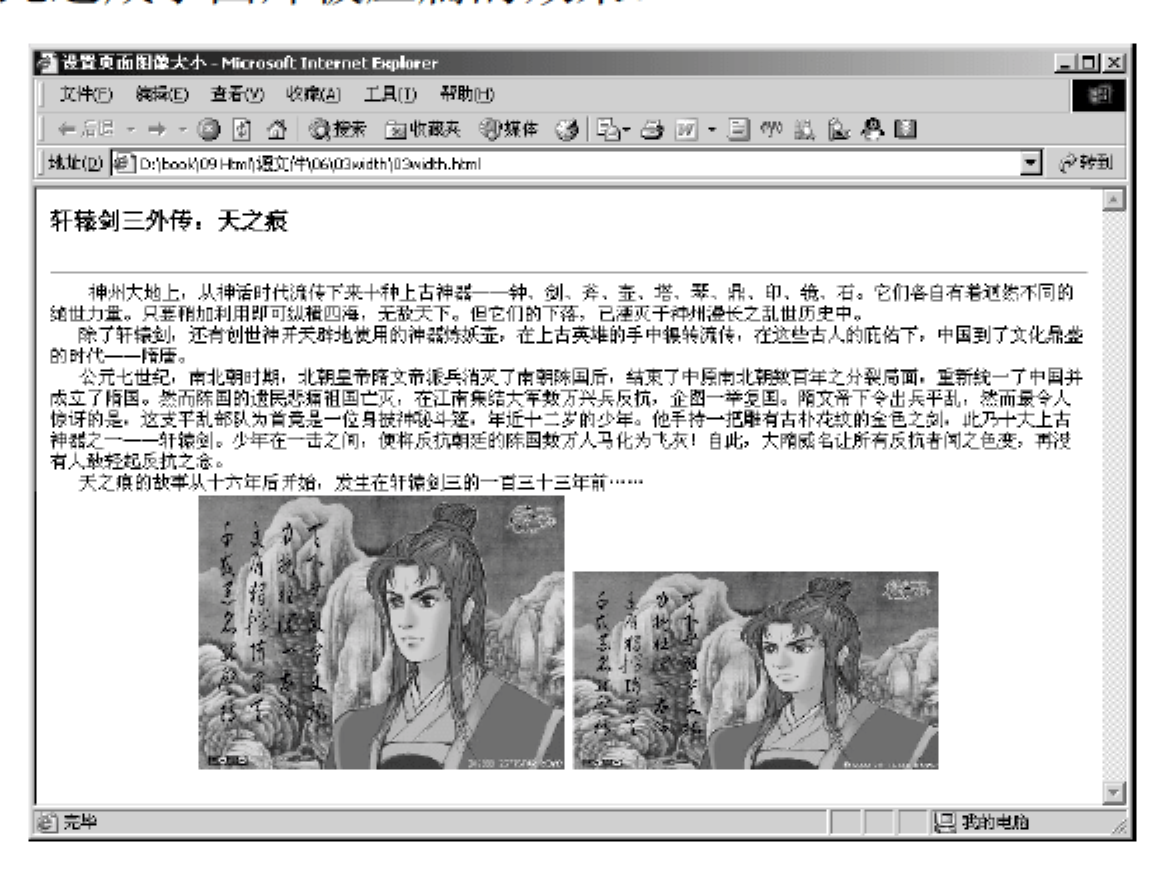


图 6-3　设置图像的大小

6.3.3　图像边框——border

在默认情况下，页面中插入的图像是没有边框的，可以通过 border 属性为图像添加边框。

语法：<img src="图像文件的地址" border="图像边框的宽度">

说明：在该语法中，src 是图像文件的地址，是不可缺少的。border 的单位是像素。

实例代码：

```
<html>
<head>
<title>设置图像边框</title>
</head>
<body>
     <h3>美女与野兽（Beauty and the Beast）</h3>
     <hr size=2>
         在一个被施了魔咒的古堡里，一位王子因不懂得慈悲与宽容而被仙女变为野兽，惟有
```

```
他学会如何爱人才能变回原貌，后来美女贝儿为救父亲而答应被野兽囚禁，在一群魔堡仆役的穿针引线之下，贝儿与野兽从针锋相对到相知相守，经历了一番波折之后，最后才终于得以解开魔法的桎梏。……
<br>
    <!--在页面中居中插入两张图片-->
    <center>
        <!--不设置图片的边框-->
        <img src="pic.jpg">
        <!--设置图片的边框为 5 像素-->
        <img src="pic.jpg" border="5">
    </center>
</body>
</html>
```

运行这段代码，效果如图 6-4 所示。可以看到在右侧图像的周围添加了边框的效果，边框的宽度为 5 像素。

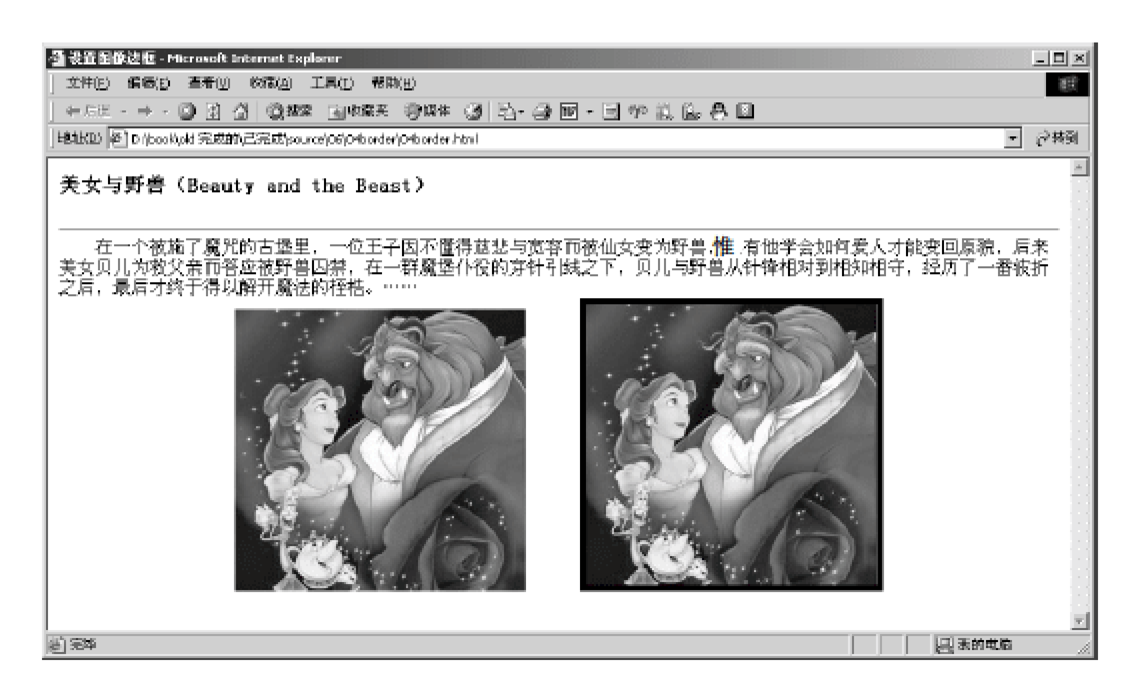


图 6-4　设置图像的边框

6.3.4　图像水平间距——hspace

如果不使用
标记或者<p>标记进行换行显示，那么添加的图像会紧跟在文字之后。而图像与文字之间的水平距离是可以通过 hspace 参数进行调整的。通过调整间距，可以使文字和图像的排版不那么拥挤，看上去更加协调。

语法：<img src="图像文件的地址" hspace="水平间距">

说明：在该语法中，src 是图像文件的地址，是不可缺少的。水平间距 hspace 属性的单位是像素。

实例代码：

```
<html>
<head>
<title>设置图像的水平间距</title>
</head>
<body>
    <h3>请选择您喜欢的头像</h3>
    <hr size=2>
    <!--在页面中居中插入两张图片-->
    <!--不设置图片水平间距的效果-->
    人物头像<img src="pic01.jpg" border="2">
```

```
            <img src="pic02.jpg" border="2">
            <img src="pic03.jpg" border="2">
            <img src="pic04.jpg" border="2">
    <br><br>
    <!--设置图片的水平间距为 20 像素-->
    动物头像<img src="pic11.jpg" border="2" hspace="20">
            <img src="pic12.jpg" border="2" hspace="20">
            <img src="pic13.jpg" border="2" hspace="20">
            <img src="pic14.jpg" border="2" hspace="20">
</body>
</html>
```

运行代码的效果如图 6-5 所示，其中人物头像一行没有设置水平间距，图片和文字紧紧连在一起；在动物头像一行设置了 20 像素的间距，图像与文字就显得不那么拥挤了。



图 6-5　设置图像的水平间距

6.3.5　图像垂直间距——vspace

垂直间距参数 vspace 用来调整图像与文字的垂直距离。

语法：<img src="图像文件的地址" vspace="垂直间距">

说明：在该语法中，vspace 属性的单位是像素。

实例代码：

```
<html>
<head>
<title>设置图像的垂直间距</title>
</head>
<body>
    <img src="pic.jpg" vspace="40">英国著名的音乐剧超级天后莎拉布莱曼的新专辑《伊甸园》隆重推出。《伊甸园》共收录了十五首歌曲，曲风涵盖了古典、跨界与流行，专辑中莎拉依旧保持她美如天籁的歌声。<br>
    英国著名的音乐剧超级天后莎拉·布莱曼的新专辑《伊甸园》隆重推出。《伊甸园》共收录了十五首歌曲，曲风涵盖了古典、跨界与流行，专辑中莎拉依旧保持她美如天籁的歌声。以她最美丽的声音来诠释这些极富古典气息的歌曲，相信喜欢莎拉·布莱曼（Sarah Brightman）的歌迷应该不会错过这张充满古典之美的专辑。音乐是她所热爱的东西，她自己的音乐也是这样，“我演唱我热爱的音乐。如果我变得不诚实，就不再成为音乐。你必须表达你内心的情感，听众知道他们是否被欺骗。”同名主打歌《伊甸园》由她亲自填词，是
```

```
她个人最喜欢的作品之一。
</body>
</html>
```

运行这段代码，效果如图 6-6 所示，为图像设置了 40 像素的垂直间距。

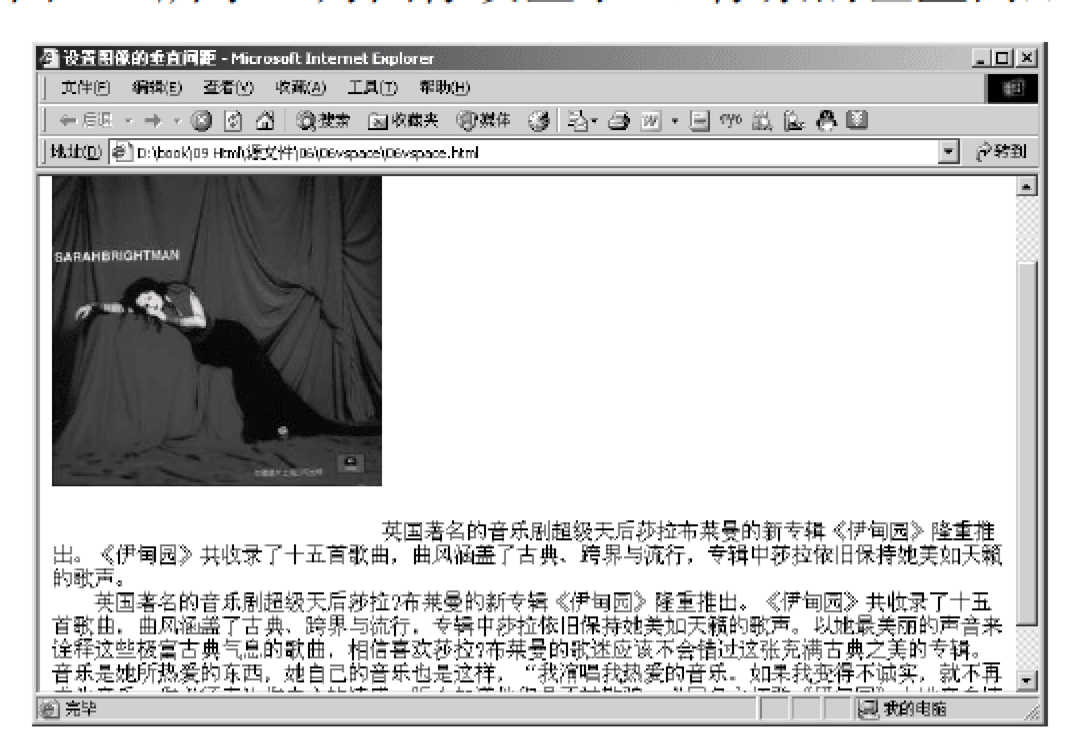


图 6-6　设置图像的垂直间距

6.3.6　图像相对于文字基准线的对齐方式——align

图像和文字之间的排列方式可以通过 align 参数来调整。图像的绝对对齐方式与相对文字的对齐方式不同，绝对对齐方式包括左对齐、右对齐和居中对齐 3 种，而相对文字对齐方式则是指图像与一行文字的相对位置。

语法：<img src="图像文件的地址" align="相对文字的对齐方式">

说明：在该语法中，align 的取值见表 6-1。

表6-1　图像相对文字的对齐方式

align取值	表示的含义
top	把图像的顶部和同行的最高部分对齐（可能是文本的顶部，也可能是图像的顶部）
middle	把图像的中部和行的中部对齐（通常是文本行的基线，并不是实际的行的中部）
bottom	把图像的底部和同行文本的底部对齐
texttop	把图像的顶部和同行中最高的文本的顶部对齐，常用于Netscape中
absmiddle	把图像的中部和同行中最大项的中部对齐，常用于Netscape中
baseline	把图像的底部和文本的基线对齐，常用于Netscape中
absbottom	把图像的底部和同行中的最低项对齐，常用于Netscape中
left	使图像和左边界对齐（文本环绕图像）
right	使图像和右边界对齐（文本环绕图像）

在表中，Netscape 支持 texttop、baseline、absmiddle 和 absbottom 的取值。

实例代码：

```
<html>
<head>
<title>设置图像与文字的相对位置</title>
```

```
</head>
<body>
    <font size=4>作为基准的文字</font>
    <!--图像的底端与文字的底端对齐-->
    <img src="pic.gif" width=85 align="bottom">
    <!--图像的中间与文字的中间线对齐-->
    <img src="pic.gif" width=85 align="middle">
    <!--图像的顶端与文字的顶端对齐-->
    <img src="pic.gif" width=85 align="texttop">
    <!--图像的中间线与文字的中间线对齐-->
    <img src="pic.gif" width=85 align="absmiddle">
    <!--图像的底端与文字的底端对齐-->
    <img src="pic.gif" width=85 align="baseline"><br>
    <hr size=2>
    下面显示图像位于文字左侧的效果：<br><br>
    <img src="pic.gif" align="left">本实例讲解的是关于图像与文字的相对位置的设置，文字作为图像对齐效果的参照物。如果将图像的对齐方式设置成 left 或者 right，图像则会与文字进行环绕显示。这是图像设置为 left 的效果。<br><br><br>
    下面显示图像位于文字右侧的效果：<br><br>
    <img src="pic.gif" align="right">本实例讲解的是关于图像与文字的相对位置的设置，文字作为图像对齐效果的参照物。如果将图像的对齐方式设置成 left 或者 right，图像则会与文字进行环绕显示。这是图像设置为 right 的效果。<br><br>
</body>
</html>
```

运行这段代码，可以看到在水平线上面是文字与图像的相对垂直位置的变化效果；而在水平线下面，则是左对齐和右对齐的效果，如图 6-7 所示。

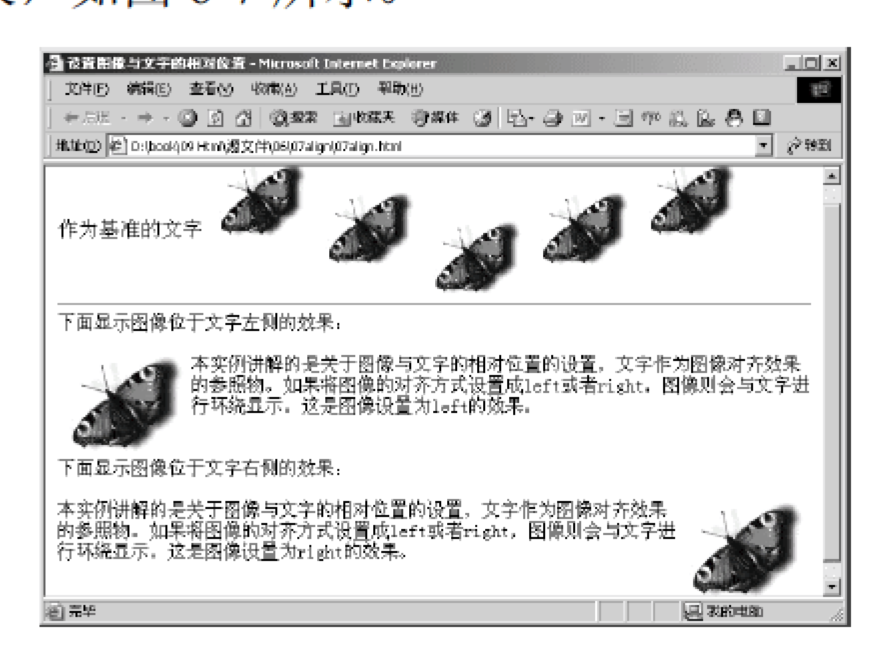


图 6-7　设置图像与文字的相对位置

6.3.7　图像的提示文字——alt

如果网络上观看 Web 站点的人使用了一个非图像化的浏览器，或者为了加快浏览器速度关掉了图像显示，这时候提示文字就可以起作用了。当图像没有装载到浏览器上时，就会显示添加的提示文字，而下载图像之后，当鼠标停留在图像上方时也会显示出提示文字，这一功能通过 alt 属性来实现。

语法：<img src="图像文字的地址" alt="提示文字的内容">

说明：在该语法中，提示文字的内容可以是中文，也可以是英文。

实例代码：

```
<html>
<head>
<title>为图像添加提示文字</title>
</head>
<body>
<h3>史密斯夫妇——Mr. and Mrs. Smith</h3>
    <img src="pic.jpg" hspace="20" align=left alt="这是 6 月 6 日发布的由美国影星布拉德皮特（右）和安吉丽娜朱莉主演的影片《史密斯夫妇》的海报。该影片将于 2005 年 6 月 10 日在全美上映。">影片介绍：布拉德皮特和安吉丽娜朱莉所扮演的史密斯夫妇在外人眼里是一对令人羡慕的夫妻，他们家境优越，夫妻恩爱。但多年的夫妻生活却多少磨平当初的激情，而多些许平淡与乏味，而这一切都仅仅是假象而已，这夫妇二人都是受过专业训练，身经百战，杀人无数的职业杀手。他们各自效力于对立的两个秘密组织，可能由于双方都是伪装技法高超的超级杀手，虽然同室而居七年有余，但各自却对对方的秘密身份知之甚少。而在一次执行刺杀任务中，夫妇二人恰好被同时指派去刺杀同一个目标，这使得二人为完成任务不得不同室操戈，而结果却是两败俱伤。为了找到导致自己任务失败的对方，夫妇俩开始调动一切资源调查对方的身份，而随着调查的深入，越来越多的证据指向同自己生活多年的爱人，因此，夫妻二人的猜疑和试探也越加深入，直到妻子在晚餐中无意将自己敏捷的身手暴露给同是内行的杀手丈夫，之后……
</body>
</html>
```

运行这段代码，当鼠标位于图像上面时可以看到添加的说明文字，如图 6-8 所示。

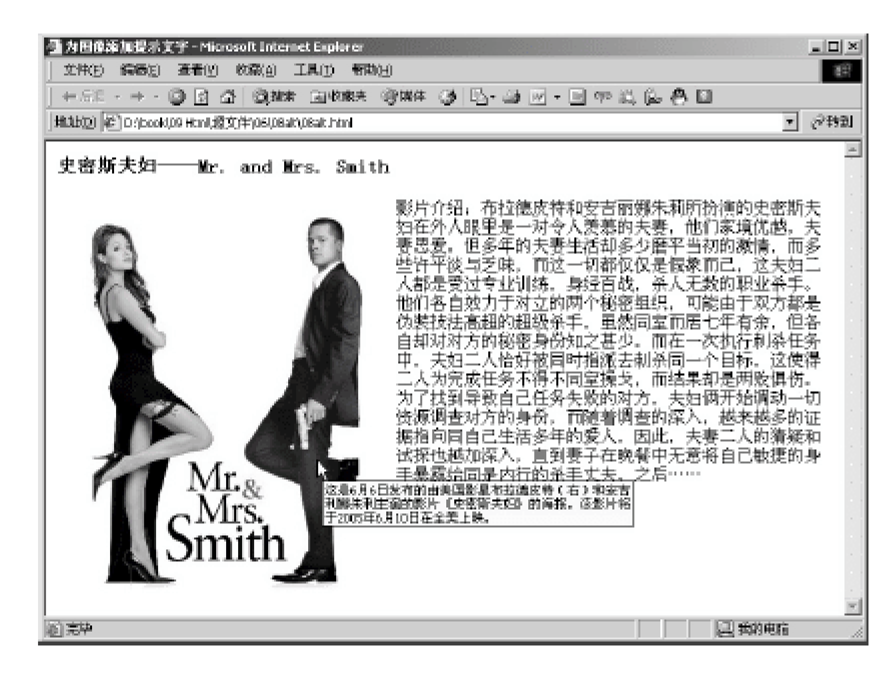


图 6-8　图像的提示文字

6.4　图像的超链接

除了文字可以添加超链接之外，图像也可以设置超链接属性。而一幅图像可以切分成不同的区域设置链接，而这些区域被称为热区。因此一幅图像也就可以设置多个链接地址。

6.4.1　设置图像的超链接

对于给整个一幅图像文件设置超链接来说，方法比较简单，与文本链接类似。

语法：<A href="链接地址" target="目标窗口的打开方式"><img src="图像文件的地址"></A>

说明：在该语法中，href 参数用来设置图像的链接地址，而在图像属性中可以添加图像的其他参

数，如 height、border、hspace 等。

下面设置一个实例，该实例的第一个页面 index.html 是一幅中国地图，单击该页面中的图像，可以链接到一个介绍中国的文字页面 china.html。

实例中 index.html 文件的代码如下：

```
<html>
<head>
<title>为图像添加超链接</title>
</head>
<body>
<h3>中国是个优秀的国家</h3>
    <center>
        <A href="china.html"><img src="map.jpg" hspace="10" alt="中国"></A>
    </center>
</body>
</html>
```

实例中 china.html 文件的代码如下：

```
<html>
<head>
<title>给图像添加超链接</title>
</head>
<body>
           中华人民共和国，简称中国，位于亚洲大陆东部，太平洋西岸。陆地面积约960 万平方公里，在亚洲，是面积最大的国家，在世界上，是仅次于俄罗斯和加拿大的第三大国家。<br>
           中国领土北起黑龙江省漠河以北的黑龙江江心，南到南沙群岛的曾母暗沙，南北相距约 5500 公里；东起黑龙江与乌苏里江的汇合处，西到帕米尔高原，东西相距约 5000 公里。陆地边界长约2.28 万公里。中国临海海域广阔，从北向南依次是渤海、黄海、东海和南海，海域面积共约 473 万平方公里。海上分布着 5400 多个岛屿。中国海岸线总长度 3.2 万公里，其中大陆海岸线 1.8 万公里，岛屿海岸线 1.4 万公里。
<br>
</body>
</html>
```

运行文件 index.html，可以看到打开的页面中包含一幅中国地图，当鼠标停留在图片上方时，会出现“中国”的提示文字，如图 6-9 所示。

在图像中单击鼠标，页面将会跳转到 china.html 文件中，效果如图 6-10 所示。

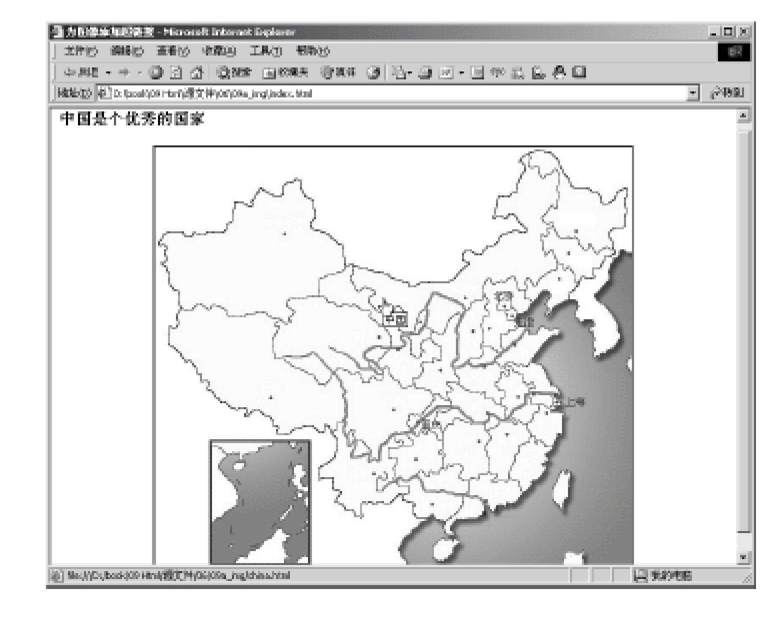


图 6-9　图像链接页面的效果

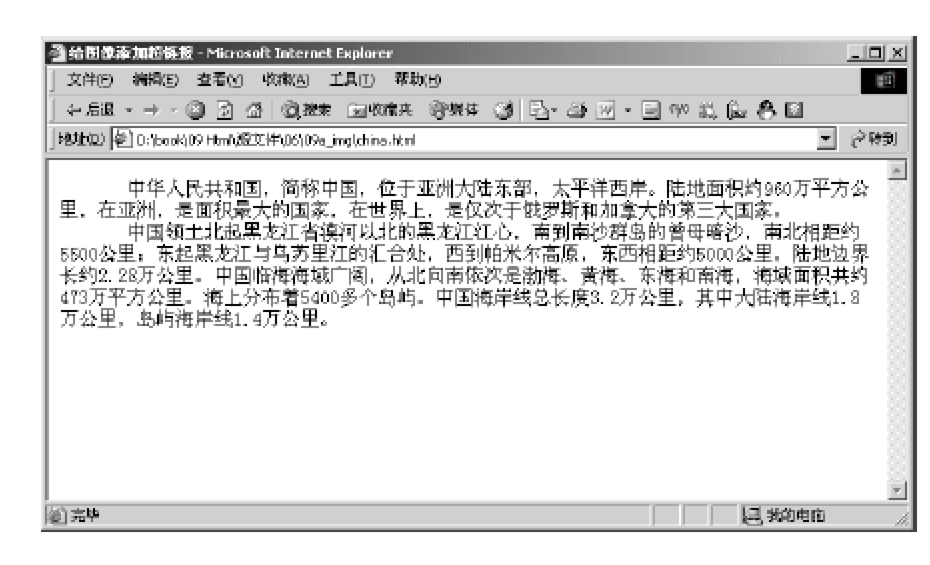


图 6-10　图像超链接的跳转页面

6.4.2 设置图像热区链接

除了对整个图像进行超链接的设置外，还可以将图像划分成不同的区域进行链接设置。而包含热区的图像也可以称为映射图像。

语法：

首先需要在图像文件中设置映射图像名，在图像的属性中使用<usemap>标记添加图像要引用的映射图像的名称如下：

```
<img src="图像地址" usemap="映射图像名称">
```

然后需要定义热区图像以及热区的链接属性如下：

```
<map name="映射图像名称">
    <area shape="热区形状" coords="热区坐标" href="链接地址">
</map>
```

说明：在该语法中要先定义映射图像的名称，然后再引用这个映射图像。在<area>标记中定义了热区的位置和链接，其中 shape 用来定义热区形状，可以取值为 rect（矩形区域）、circle（圆形区域）以及 poly（多边形区域）；coords 参数则用来设置区域坐标，对于不同形状来说，coords 设置的方式也不同。

对于矩形区域 rect 来说，coords 包含 4 个参数，分别为 left、top、right 和 bottom，也可以将这 4 个参数看作矩形两个对角的点坐标；对于圆形区域 circle 来说，coords 包含 3 个参数，分别为 center-x、center-y 和 tadius，也可以看作是圆形的圆心坐标（x，y）与半径的值；对于多边形区域 poly 设置坐标参数比较复杂，跟多边形的形状息息相关。coords 参数需要按照顺序（可以是逆时针，也可以是顺时针）取各个点的 x、y 坐标值。

由于定义坐标比较复杂而且难以控制，一般情况下都使用可视化软件进行这种参数的设置。具体方法如下：

（1）编写一个 HTML 文件，代码如下：

```
<html>
<head>
<title>给图像添加超链接</title>
</head>
<body>
<h3>中国是个优秀的国家</h3>
    <center>
        <img src="map.jpg" hspace="10"><br><br>
    </center>
    <hr color="#990000" size=3><br><br>
           中华人民共和国，简称中国，位于亚洲大陆东部，太平洋西岸。陆地面积约
960 万平方公里，在亚洲，是面积最大的国家，在世界上，是仅次于俄罗斯和加拿大的第三大国家。<br>
           中国领土北起黑龙江省漠河以北的黑龙江江心，南到南沙群岛的曾母暗沙，南
北相距约 5500 公里；东起黑龙江与乌苏里江的汇合处，西到帕米尔高原，东西相距约 5000 公里。陆地边界长约
2.28 万公里。中国临海海域广阔，从北向南依次是渤海、黄海、东海和南海，海域面积共约 473 万平方公里。海
上分布着 5400 多个岛屿。中国海岸线总长度 3.2 万公里，其中大陆海岸线 1.8 万公里，岛屿海岸线 1.4 万公里。
<br>
           中国行政区包括省、直辖市、自治区以及特别行政区。直辖市的行政地位相当
于省级，目前我国直辖市的数目为 4 个，分别为北京、上海、天津、重庆。<br><br>
    <font color="#990066" size=4><a name="beijing">北京</a></font><br>
            北京是我国的首都，全国的政治、文化中心和国际交往的枢纽，也是一座著
```

```
名的“历史文化名城”。北京位于华北平原的最北端，西与黄土高原相邻，北与内蒙古高原相接，处于高原相接、山地和平原衔接部。北京地处温带半干旱、半湿润季风气候区，四季分明。北京有着大气，严肃，正统而又不失闲适、清雅的文化氛围。<br><br>
    <font color="#990066" size=4><a name="tianjin">天津</a></font><br>
            天津市简称津，地处华北平原东北部，渤海之滨，素有渤海明珠之称。天津毗邻首都北京，是海上通往北京的咽喉要道，自古就是京师门户，畿辅重镇。天津是华北的一大工业城市，是华北重要商业中心和口岸城市，油气、海盐资源丰富，又有一定的工业技术基础。天津的旅游景点以人文景观为主，以自然景观为辅，主要由三个部分组成：以海河为轴线的市中心旅游区，以名胜古迹见长的蓟县旅游区和以滨海观光为特色的塘沽旅游区。<br><br>
    <font color="#990066" size=4><a name="shanghai">上海</a></font><br>
            上海地处长江三角洲前沿，倚东海之滨，向东是浩瀚无垠的太平洋、与美国的西海岸隔海相望，南临杭州湾，西与富庶的江苏、浙江两省毗邻，北界黄金水道长江入海口，正当我国南北海岸线的中部，交通便利，地理位置十分优越，是世界第三大港和中国最大的港口。上海虽然尚没有雄伟的名山大川、奇峰异谷，也无世界奇迹之类的名胜古迹，但是，多少年来一直以她独有的风韵吸引着无数的中外游客。她是一个不断发展日渐强盛的城市，是我国最大的商业、金融中心，也是西太平洋地区重要的国际港口城市。<br><br>
    <font color="#990066" size=4><a name="chongqing">重庆</a></font><br>
            重庆位于青藏高原与长江中下游平原的过渡地带。气候属亚热带季风性湿润气候，冬暖夏热，无霜期长、雨量充沛、温润多阴、雨热同季。重庆地处我国中西结合部，是承东启西、左右传递的枢纽，在我国经济发展总格局和西部大开发中，具有重要的战略地位和作用。重庆中心城区为长江、嘉陵江所环抱，夹两江、拥群山，山清水秀，风景独特，各类建筑依山傍水，鳞次栉比，错落有致，素以美丽的“山城”、“江城”著称于世。北有大巴山，东有巫山，东南有武陵山，南有大娄山，地形大势由南北向长江河谷倾斜，起伏较大。地貌以丘陵、山地为主，坡地面积较大，成层性明显，分布着典型的石林、峰林、溶洞、峡谷等喀斯特景观。
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>
</body>
</html>
```

（2）启动 Dreamweaver MX 2004，选择“文件”|“打开”命令，弹出“打开”对话框，在对话框中选择刚才编写的 HTML 文件，单击“打开”按钮打开 HTML 文件，效果如图 6-11 所示。

（3）在属性面板中设置“地图”的值为 map，也就是定义图像要引用的映射图像名。

（4）为了显示各种热区的效果，这里将不同的热区设置为不同的形状。选择“矩形热点工具”□，拖动鼠标左键在北京的位置上绘制出一个矩形区域，如图 6-12 所示。



图 6-11　打开的文件

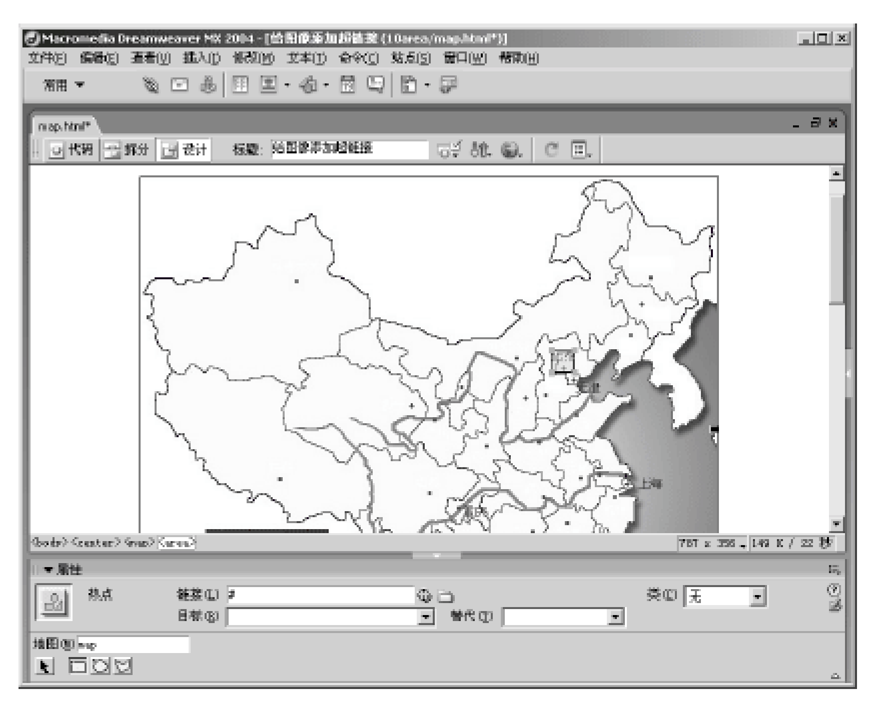


图 6-12　绘制矩形区域

（5）在“链接”文本框中添加链接地址“#beijing”，在“替代”文本框设置选区的提示文字为“北京”。

（6）再次单击图片，选择“多边形热点工具”，依次单击热区的各个点创建一个多边形选区，如图 6-13 所示。

（7）在属性面板中设置链接地址为“#tianjin”，设置提示文字为“天津”。

（8）用同样的方法设置两个椭圆形选区，分别设置链接地址为“#shanghai”和“#chongqing”；然后设置对应的提示文字为“上海”和“重庆”。

（9）完成后保存文件，使用浏览器运行这个文件。当鼠标位于某一个热区上方时，会出现该热区的提示文字与链接标志，效果如图 6-14 所示。

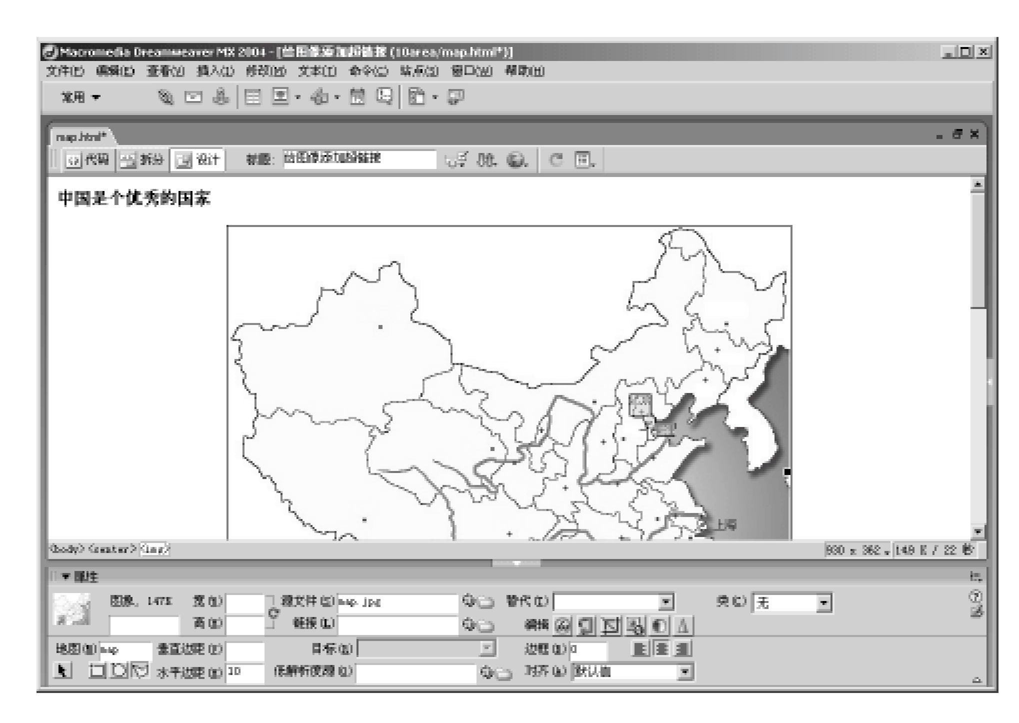


图 6-13　绘制多边形选区

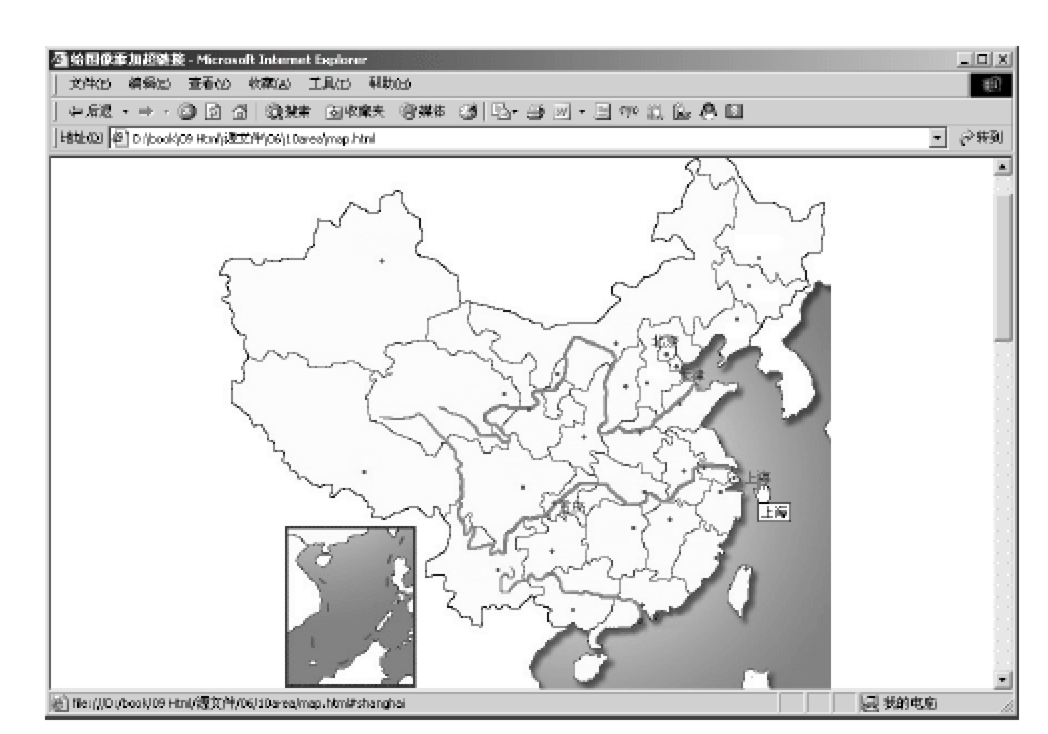


图 6-14　运行文件的效果

单击上海所在的热区，将会自动跳转到关于上海的介绍文字的位置，如图 6-15 所示。运行其他热区的效果类似，这里不再列举。

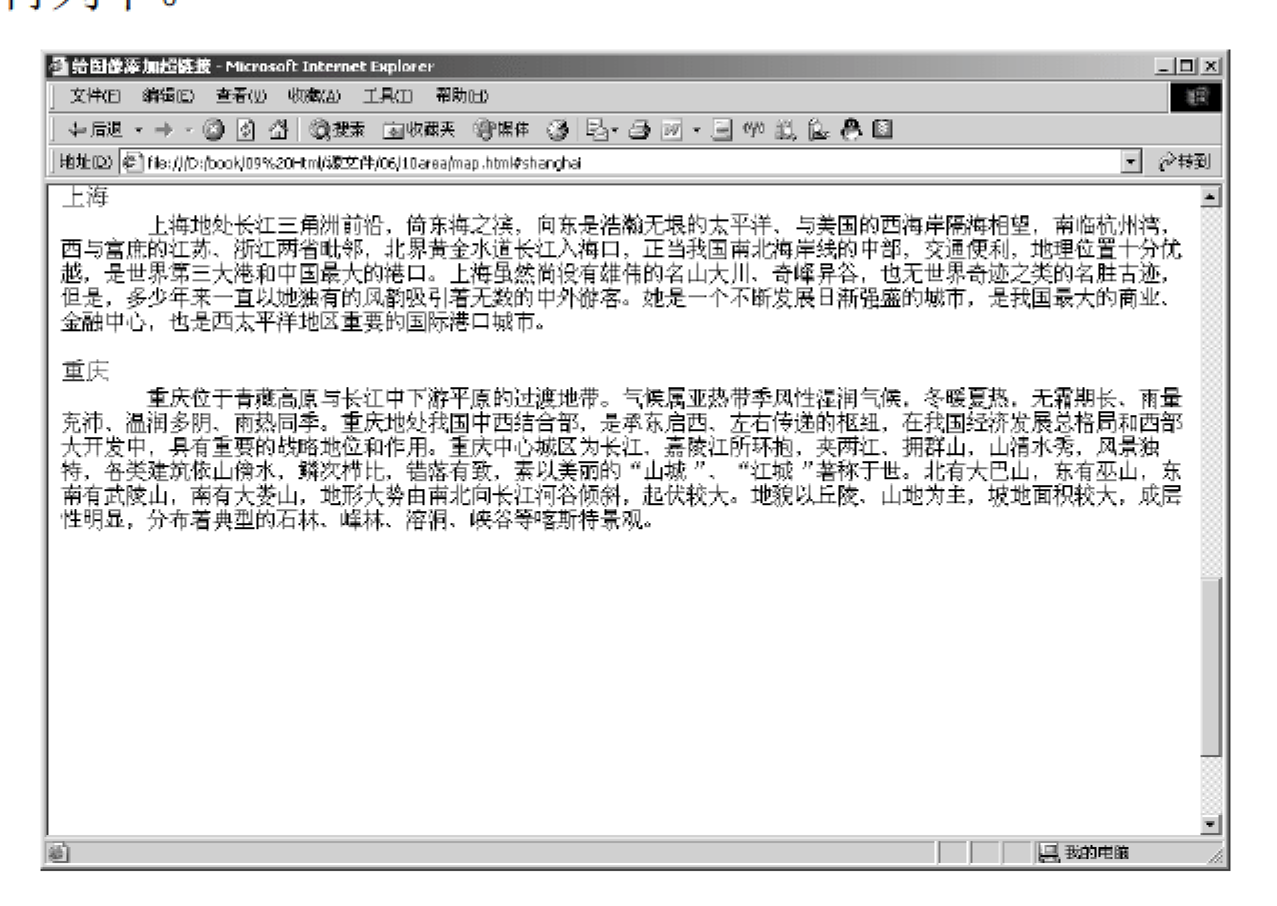


图 6-15　跳转到热区的链接地址

重新打开源文件，可以看到在源文件添加图像的代码中增加了部分内容。在原来的文件代码中插入图像的代码如下：

```
<img src="map.jpg" hspace="10"><br><br>
```

这是位于标记<center>与</center>之间的内容，而增加后的代码如下：

```
<img src="map.jpg" hspace="10" border="0" usemap="#map">
<map name="map">
        <area shape="rect" coords="398,168,420,190" href="#beijing" alt="北京">
        <areashape="poly"coords="413,194,419,188,423,192,423,197,443,197,442,209,422,208,414,201"
href="#tianjin" alt="天津">
      <area shape="circle" coords="317,323,17" href="#chongqing" alt="重庆">
      <area shape="circle" coords="485,299,17" href="#shanghai" alt="上海">
</map>
<br><br>
```

这段代码就是设置热区的关键代码，在这段代码中设置了 4 个不同形状的热区，并分别为这些热区设定了链接地址和提示文字。

第 7 章

添加多媒体元素

- 设置动态文字
- 设置背景音乐
- 添加多媒体文件

多媒体是一个网站的必备元素，使用它可以丰富网站效果，体现设计者的个性，吸引用户的注意，突出重点。通常多媒体元素包括声音和动画两部分，本章将详细介绍。

7.1　设置动态文字

网页的多媒体元素一般包括动态文字、动态图像、声音以及动画等，其中最简单的就是添加一些滚动文字。

7.1.1　设置滚动文字——marquee

使用 marquee 标记可以将文字设置为动态滚动的效果。

语法：<marquee>滚动文字</marquee>

说明：只要在标记之间添加要进行滚动的文字即可，而且可以在标记之间设置这些文字的字体、颜色等。

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
	<marquee>
		<font face="隶书" color="#CC0000 " size=4>你好，欢迎光临梦幻小屋!这里有欢乐的歌声，这里有美好
的景色</font>
	</marquee>
</body>
</html>
```

运行这段代码，可以看到设置为红色隶书的文字从浏览器的右方缓缓向左滚动，如图 7-1 所示。

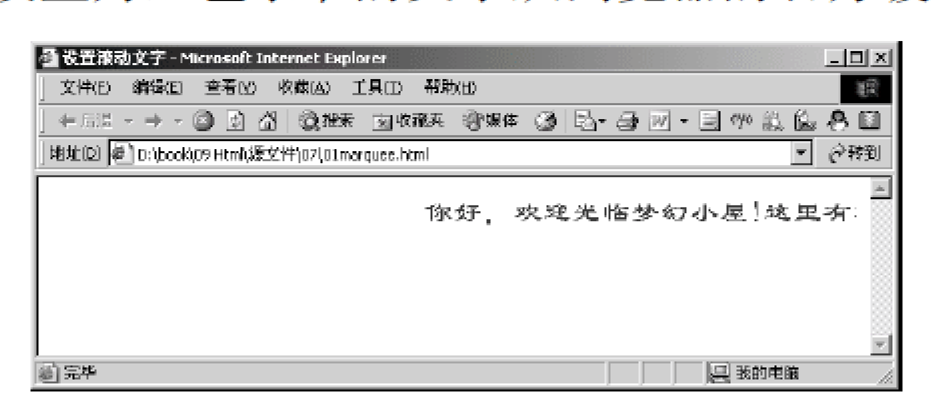


图 7-1　设置滚动文字

7.1.2　文字滚动方向——direction

默认情况下文字只能是从右向左滚动，而在实际应用中常常需要不同滚动方向的文字，这可以通过 direction 参数来设置。

语法：<marquee direction="滚动方向">滚动文字</marquee>

说明：该语法中的滚动方向可以包含 4 个取值，分别为 up、down、left 和 right，它们分别表示文字向上、向下、向左和向右滚动，其中向左滚动 left 的效果与默认效果相同，而向上滚动的文字则常

常出现在网站的公告栏中。

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
	<marquee direction="up">
		<font color="#3300FF" face="楷体_GB2312">
			你好，欢迎您的光临<br>
			这里是梦想小屋<br>
			让我们与您分享您的点点快乐<br>
			让我们与您分担您的片片忧伤<br>
		</font>
	</marquee>
	<marquee direction="down">
		<font color="#FF0000" face="楷体_GB2312">
			你好，欢迎您的光临<br>
			这里是梦想小屋<br>
			让我们与您分享您的点点快乐<br>
			让我们与您分担您的片片忧伤<br>
		</font>
	</marquee>
</body>
</html>
```

运行这段代码，可以看到文字不同的滚动效果，如图 7-2 和图 7-3 所示。图中两段文字的感觉就像是从中间被拉扯开一样。



图 7-2　设置滚动方向



图 7-3　运行一段时间的效果

如果将文字方向相反设置，也就是上面一段文字的 direction 设置为 down，下面一段设置为 up，运行结果就似乎是两组文字融合到一起一样。

7.1.3　设置文字的滚动方式——behavior

除了将文字设置为单方向的滚动外，还可以为文字设置滚动方式，如往复运动等。这一功能可以通过添加 behavior 属性来实现。

语法：<marquee behavior="滚动方式 ">滚动文字</marquee>

说明：在这里，滚动方式 behavior 的取值可以设置为表 7-1 中所示的某个值，不同取值的滚动效果也不相同。

表7-1　滚动方式的设置

behavior的取值	滚动的效果
scroll	循环滚动，默认效果
slide	只滚动一次就停止
alternate	来回交替进行滚动

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
      <marquee behavior="scroll">你好，欢迎您的光临</marquee>
      <br><br>
      <marquee behavior ="slide">让我们与您分享您的点点快乐</marquee>
      <br><br>
      <marquee behavior ="alternate">让我们与您分担您的片片忧伤</marquee>
</body>
</html>
```

运行这段代码，可以看到如图 7-4 所示的效果。其中第一行文字不停地循环，一圈一圈地滚动；而第二行文字则在第一次到达浏览器边缘时就停止了滚动；最后一行文字则在滚动到浏览器左边缘后开始反方向运动。

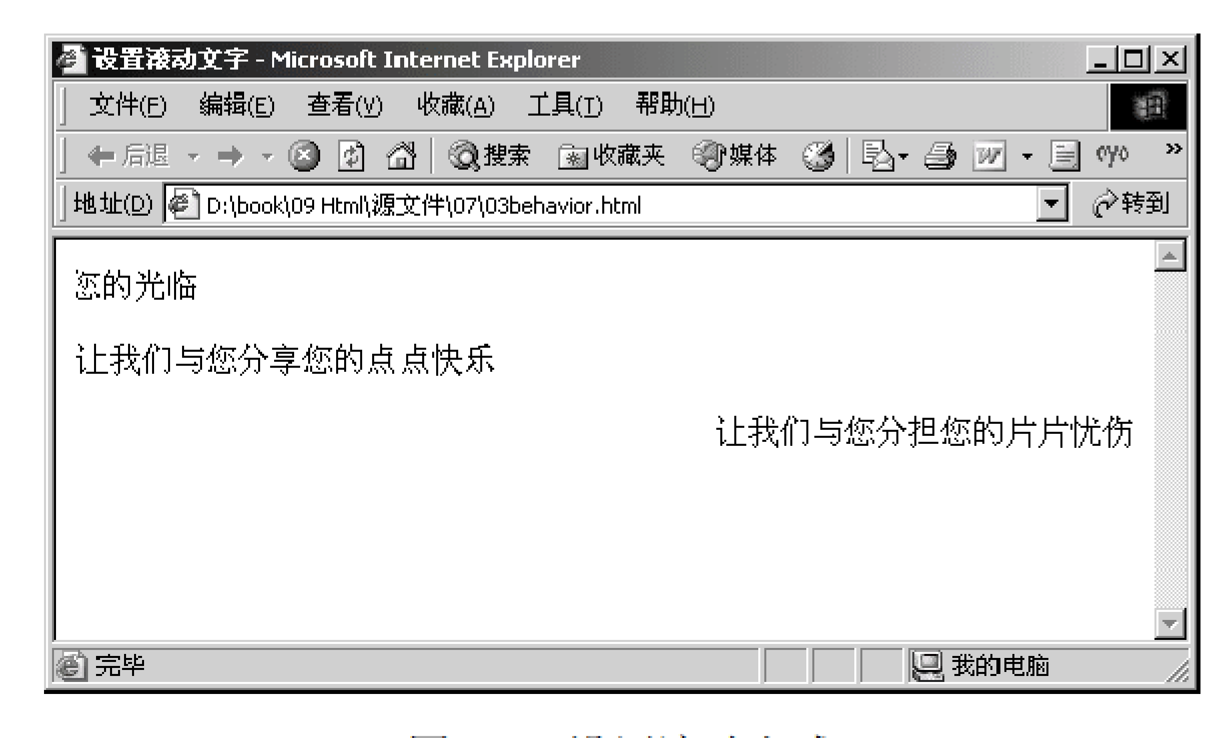


图 7-4　设置滚动方式

7.1.4　循环设置——loop

设置滚动文字后，在默认情况下会不断地循环下去，如果希望文字滚动几次停止，可以使用 loop 参数来进行设置。

语法：<marquee loop="循环次数">滚动文字</marquee>

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
    <marquee direction="up" loop="3">
        <font color="#3300FF" face="楷体_GB2312">
            你好，欢迎您的光临<br>
            这里是梦想小屋<br>
            让我们与您分享您的点点快乐<br>
            让我们与您分担您的片片忧伤<br>
        </font>
    </marquee>
</body>
</html>
```

运行这段代码，会发现当文字滚动 3 个循环之后，滚动文字将不再出现，如图 7-5 所示。但是如果设置滚动方式为交替滚动，那么在滚动 3 个循环之后，文字将停留在窗口中，如图 7-6 所示。



图 7-5　设置循环次数的效果

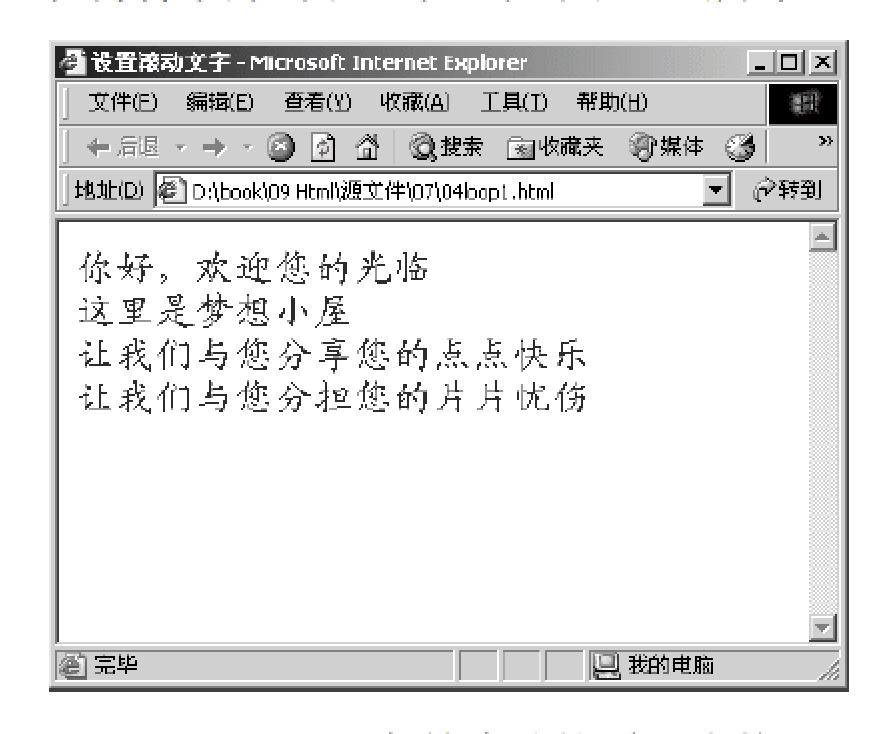


图 7-6　设置交替滚动的循环次数

7.1.5　滚动速度——scrollamount

在设置滚动文字时，有时候可能希望它快一些，也有时候希望它慢一些。这一功能可以使用 scrollamount 参数来实现。

语法：<marquee scrollamount=滚动速度></marquee>

说明：在该语法中，滚动文字的速度实际上是设置滚动文字每次移动的长度，以像素为单位。

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
    <marquee scrollamount=3>看看我走得速度怎么样！</marquee><br><br>
    <marquee scrollamount=10>看看我走得速度怎么样！</marquee><br><br>
    <marquee scrollamount=50>看看我走得速度怎么样！</marquee>
</body>
</html>
```

运行这段代码，可以看到 3 行文字同时开始滚动，但是速度是不一样的，设置的 scrollamount 越大，速度也就越快，如图 7-7 所示。

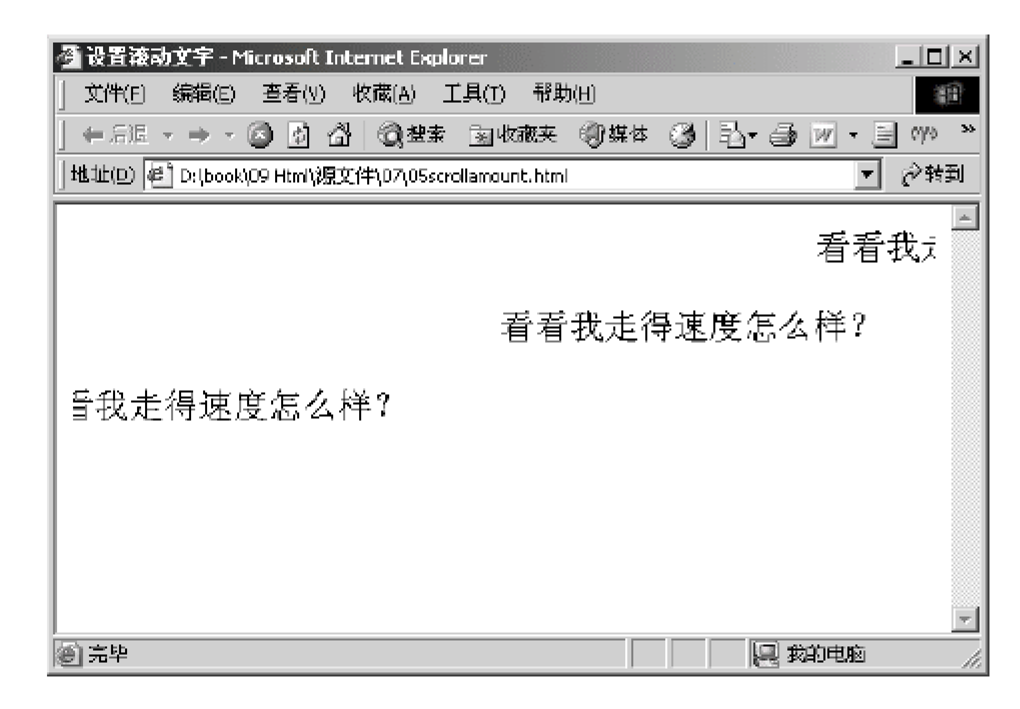


图 7-7　设置不同的滚动速度

7.1.6　滚动延迟——scrolldelay

scrolldelay 参数可以设置滚动文字滚动的时间间隔。

语法：<marquee scrolldelay=时间间隔></marquee>

说明：scrolldelay 的时间间隔单位是毫秒，也就是千分之一秒。这一时间间隔的设置为滚动两步之间的时间间隔，如果设置的时间比较长，会产生走走停停的效果。

如果与滚动速度 scrollamount 参数结合使用，效果更明显，下面以实例说明。

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
    <marquee scrollamount=100 scrolldelay =10>看我不停脚步得走！</marquee><br><br>
    <marquee scrollamount=100 scrolldelay =100>看我走走歇歇！</marquee><br><br>
    <marquee scrollamount=100 scrolldelay =500>我要走一步停一停</marquee>
</body>
</html>
```

运行这段代码，效果如图 7-8 所示，其中第一行文字设置的延迟小，因此走起来比较平滑；最后一行设置的延迟比较大，看上去就像是走一步歇一会儿的感觉。

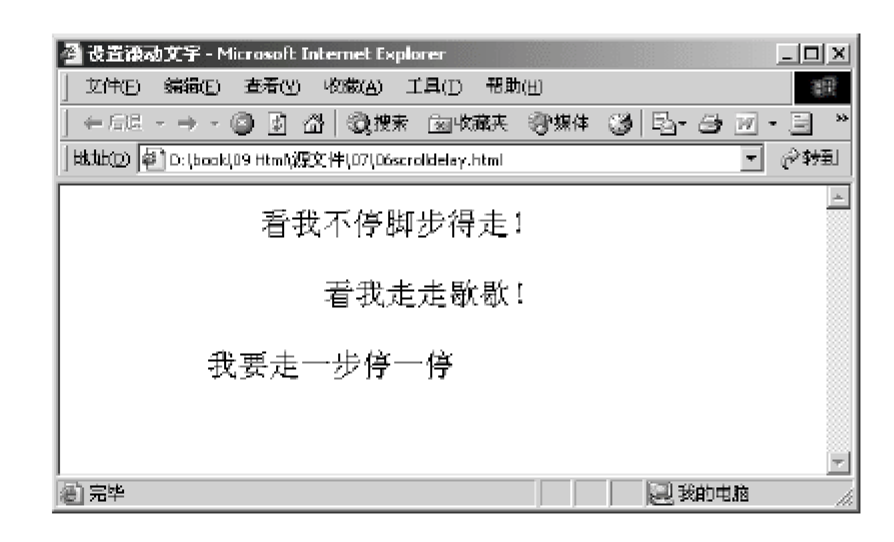


图 7-8　设置滚动延迟

7.1.7　滚动文字的背景设置——bgcolor

在网页中，为了突出某部分内容，常常使用不同背景色来显示。滚动文字也可以单独设置背景色。

语法：<marquee bgcolor="颜色代码">滚动文字</marquee>

说明：文字背景颜色设置为 16 位颜色码。

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
	<marquee behavior ="alternate" bgcolor="#FFFF66">
		这里是梦幻小屋，欢迎光临
	</marquee>
	<br><br>
	<marquee direction="up" bgcolor="#99CCFF">
			你好，欢迎您的光临<br>
			这里是梦想小屋<br>
			让我们与您分享您的点点快乐<br>
			让我们与您分担您的片片忧伤<br>
	</marquee>
</body>
</html>
```

运行这段代码，看到在滚动文字后面设置了淡蓝色的背景，如图 7-9 所示。

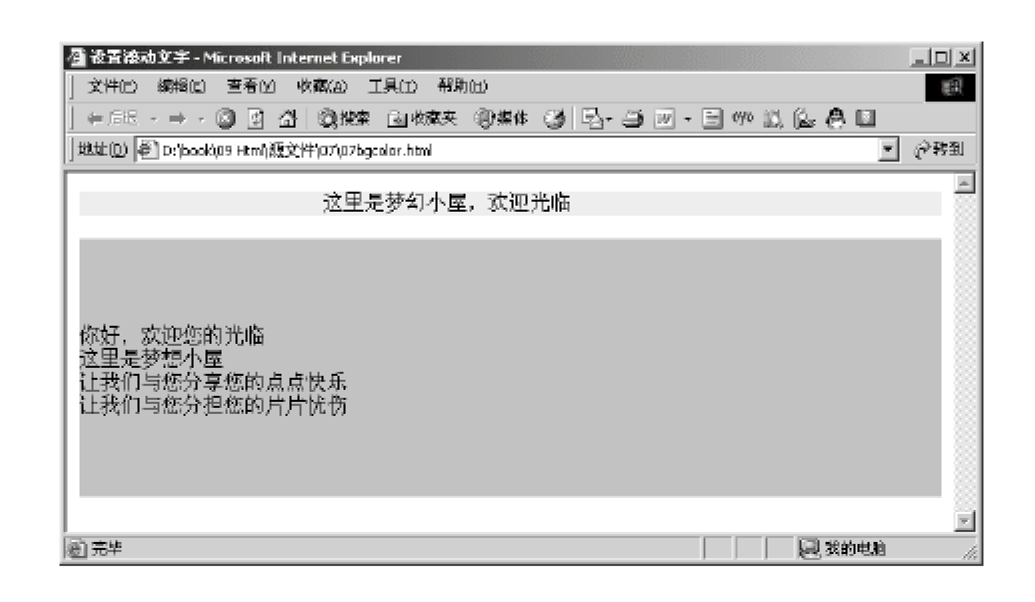


图 7-9　设置滚动文字背景

7.1.8　滚动背景面积——width、height

如果不设置滚动背景的面积，那么默认情况下，水平滚动的文字背景与文字同高、与浏览器窗口同宽，使用 width 和 height 参数可以调整其水平和垂直的范围。

语法：<marquee width=背景宽度 height=背景高度>滚动文字</maruquee>

说明：此处设置宽度和高度的单位均为像素。

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
    <marquee behavior =" alternate" bgcolor="#99CCFF">
        这里是梦幻小屋，欢迎光临
    </marquee><br><br>
    <marquee behavior="alternate"bgcolor="#99CCFF" width=500
height=50>
        这里是梦幻小屋，欢迎光临
    </marquee>
</body>
</html>
```

运行这段代码，可以看到两段滚动文字的背景高度和宽度的变化，如图 7-10 所示。

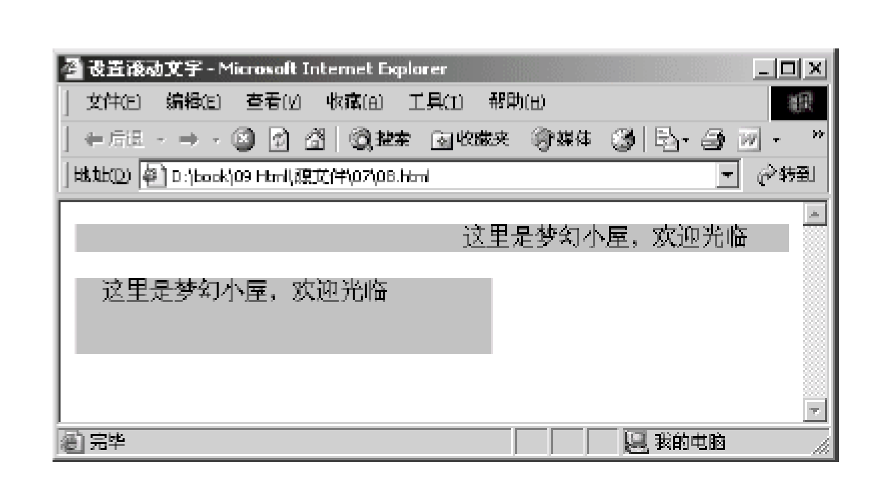


图 7-10　设置滚动文字背景的面积

7.1.9　设置空白空间——hspace、vspace

默认情况下，滚动文字周围的文字或图像是与滚动背景紧密连接的，使用参数 hspace 和 vspace 可以设置它们之间的空白空间。

语法：<marquee hspace=水平范围 vspace=垂直范围>滚动文字</marquee>

说明：该语法中水平和垂直范围的单位均为像素。

实例代码：

```
<html>
<head>
<title>设置滚动文字</title>
</head>
<body>
    不设置空白空间的效果：
    <marquee behavior ="alternate" bgcolor="#9999FF ">
        这里是梦幻小屋，欢迎光临
    </marquee>
    到这里，留下你的忧伤，带走我的快乐！
    <br>
    <hr color="#FF0000">
    <br>
    设置水平为 70 像素、垂直为 50 像素的空白空间：
    <marquee behavior ="alternate" bgcolor="#9999FF " hspace=70 vspace=50>
        这里是梦幻小屋，欢迎光临
    </marquee>
```

```
        我的梦想与你同在！
</body>
</html>
```

运行这段代码，可以看到设置空白空间的效果如图 7-11 所示。

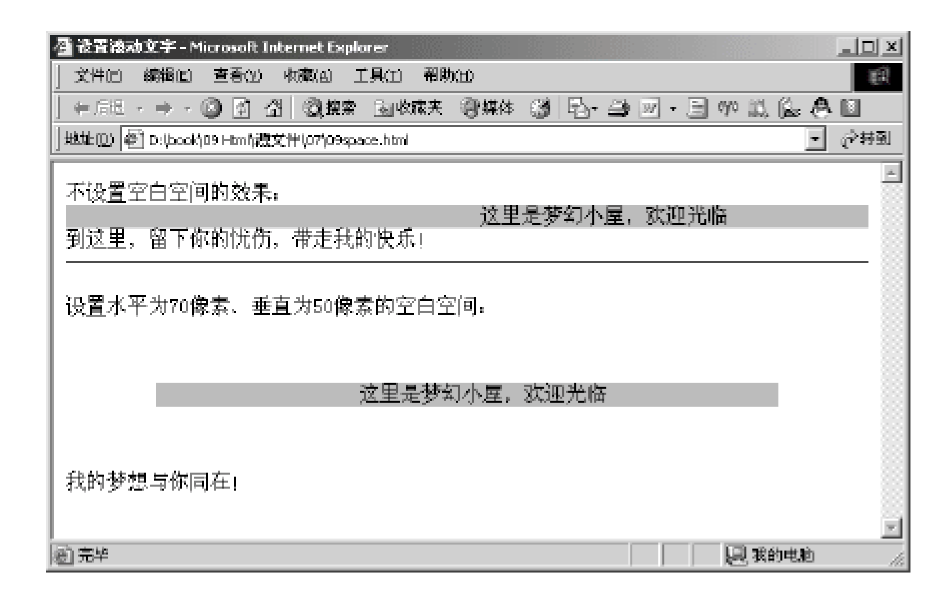


图 7-11　设置滚动文字周围的空白空间

7.2　设置背景音乐

7.2.1　设置背景音乐——bgsound

在网页中，除了可以嵌入普通的声音文件外，还可以为某个网页设置背景音乐。作为背景音乐的可以是音乐文件，也可以是声音文件，其中最常用的是 midi 文件。

语法：<bgsound src=背景音乐的地址>

说明：作为背景音乐的文件还可以是 avi 文件、mp3 文件等声音文件。

实例代码：

```
<html>
<head>
<title>设置背景音乐</title>
</head>
<body>
        <bgsound src="exam02.mid">
        <center>
                <font face="黑体" size=5>醉花阴</font><br>
                <font size=4>李清照</font>
        </center>
        <hr width=85% size=3>
        <p>    薄雾浓云愁永画，瑞脑消金兽。<br>
            佳节又重阳，玉枕纱厨，半夜凉初透。<br>
            东离把酒黄昏后，有暗香盈袖。<br>
            莫道不消魂，帘卷西风，人比黄花瘦。</p>
        <hr size=5>
</body>
</html>
```

运行这段代码，打开如图 7-12 所示的界面，在打开界面之后就可以听到背景音乐。

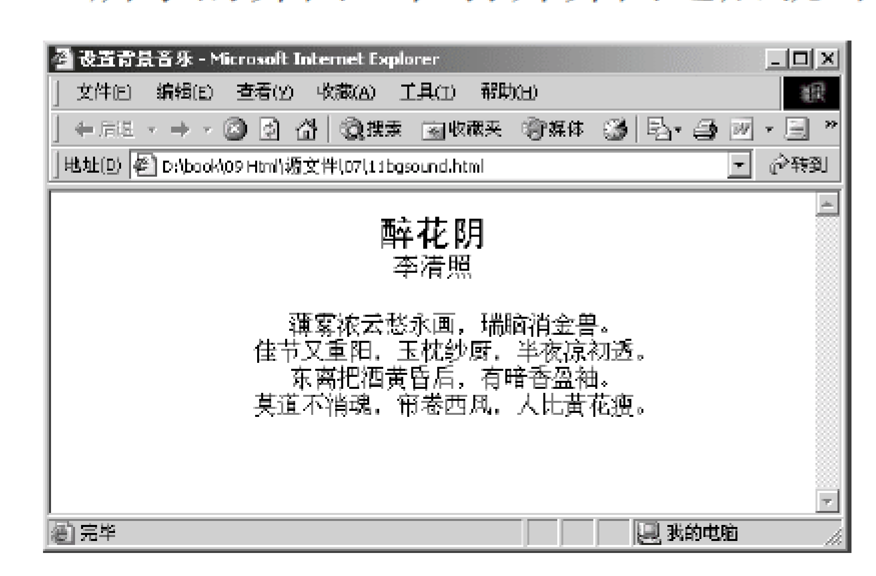


图 7-12　设置背景音乐

7.2.2　设置背景音乐的循环播放次数——loop

通常情况下，背景音乐需要不断地播放，但有时也需要指定播放的次数。这一功能实现起来并不难，只要设置相应的 loop 参数即可。

语法：<bgsound src=背景音乐的地址 loop=循环次数>

说明：如果希望无限次循环播放背景音乐，可将循环次数设置为 true。

实例代码：

```
<html>
<head>
<title>设置背景音乐</title>
</head>
<body>
	<bgsound src="exam02.mid" loop=3>
	<center>
		<font face="黑体" size=5>醉花阴</font><br>
		<font size=4>李清照</font>
	</center>
	<hr width=85% size=3>
	<p>    薄雾浓云愁永画，瑞脑消金兽。<br>
	    佳节又重阳，玉枕纱厨，半夜凉初透。<br>
	    东离把酒黄昏后，有暗香盈袖。<br>
	    莫道不消魂，帘卷西风，人比黄花瘦。</p>
	<hr size=5>
</body>
</html>
```

运行这段代码，背景音乐在循环播放 3 次以后停止。

7.3　添加多媒体文件

如果能在网页中加入音乐或视频文件，可以使单调的网页变得更加生动。但是如果要正确浏览嵌

入了这些文件的网页，就需要在客户端的计算机中安装相应的播放软件。使用<embed>标记可以将多媒体文件嵌入到网页中。

7.3.1　添加多媒体文件标记——embed

在网页中常见的多媒体文件包括声音文件和视频文件。

语法：<embed src="多媒体文件地址" width=播放界面的宽度 height=播放界面的高度></embed>

说明：在该语法中，width 和 height 一定要设置，单位是像素，否则无法正确显示播放多媒体文件的软件。

实例代码：

```
<html>
<head>
<title>嵌入多媒体文件</title>
</head>
<body>
    下面是嵌入的多媒体文件：<br>
    <embedsrc="exam01.mid"width=500height=200></embed>
</body>
</html>
```

运行这段代码，可以看到一个播放页面，如图 7-13 所示。单击页面中的播放按钮▶可以播放插入的声音文件 exam01.mid。

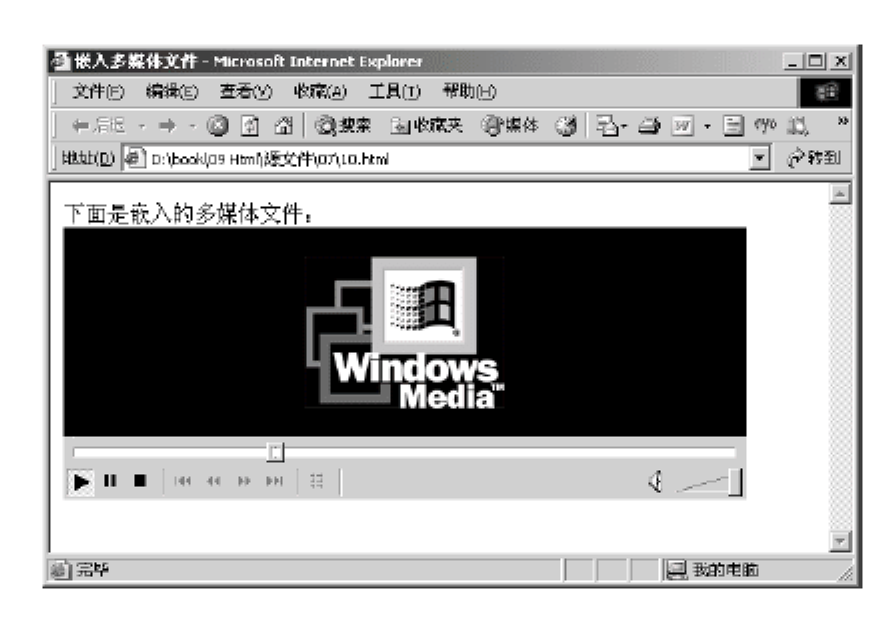


图 7-13　插入多媒体文件

7.3.2　设置自动运行——autostart

登录网页时常常会看到一些视频文件直接开始运行，不需要手动开始，特别是一些广告内容，这是通过 autostart 参数来实现的。

语法：<embed src="多媒体文件地址" autostart=是否自动运行></embed>

说明：autostart 的取值有两个：一个是 true，表示自动播放；另一个是 false，表示不自动播放。

实例代码：

```
<html>
<head>
<title>设置自动运行</title>
</head>
<body>
    下面的视频文件中左边的视频文件将会自动播放，而右边的视频文件则需要手动播放：<br>
    <embedsrc="exam01.avi"width=300 autostart=True></embed>
    <embedsrc="exam01.avi"width=300 autostart=False></embed>
</body>
</html>
```

运行这段代码，可以看到两个视频文件的不同效果，如图 7-14 所示。

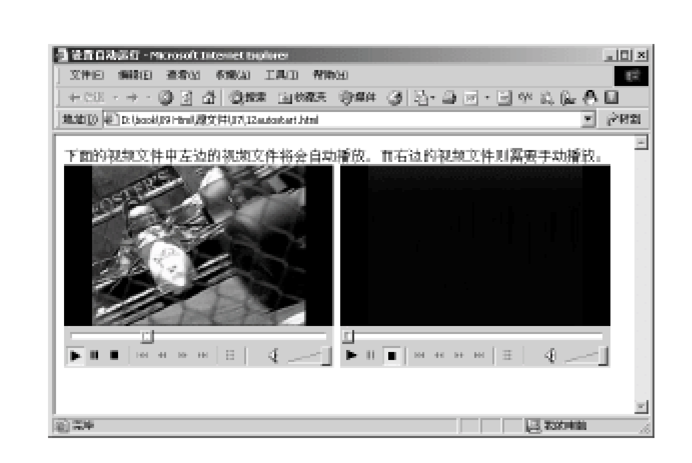


图 7-14　设置自动运行

7.3.3　设置媒体文件的循环播放——loop

这里的循环播放一般在设置了自动播放时采用，与背景音乐的设置基本相同。

语法：<embed src="多媒体文件地址" loop=是否循环播放></embed>

说明：在该语法中，loop 的取值不是具体的数字，而是 true 或者 false，如果取值为 true，表示媒体文件将无限次地循环播放；如果取值为 false，则只播放一次。这里的 loop 也可以设置为播放次数，但是用途并不广泛。

实例代码：

```
<html>
<head>
<title>设置循环播放</title>
</head>
<body>
    下面的视频文件将循环播放：<br>
    <embed src="exam01.avi" width=300 autostart=True loop=True
></embed>
</body>
</html>
```

运行这段代码，效果如图 7-15 所示。



图 7-15　媒体文件不停地循环播放

7.3.4　隐藏面板——hidden

其实也可以将媒体文件的声音保留而隐藏图像，这样就相当于设置了背景声音。通过 hidden 参数可以隐藏播放面板。

语法：<embed src="多媒体文件地址" hidden=是否隐藏></embed>

说明：在该语法中，hidden 可以设置两个值：一个是 true，表示隐藏面板；另一个是 false，表示显示面板，这是添加媒体文件的默认选项。如果要保留声音，就要设置文件的自动播放。

实例代码：

```
<html>
```

```
<head>
<title>设置隐藏面板</title>
</head>
<body>
    下面的视频文件播放面板被隐藏：<br>
    <embed src="exam01.avi"width=300 autostart=True hidden=True
></embed>
</body>
</html>
```

运行这段代码，看到播放控制面板已经不见了，只能听到播放的声音效果，如图 7-16 所示。

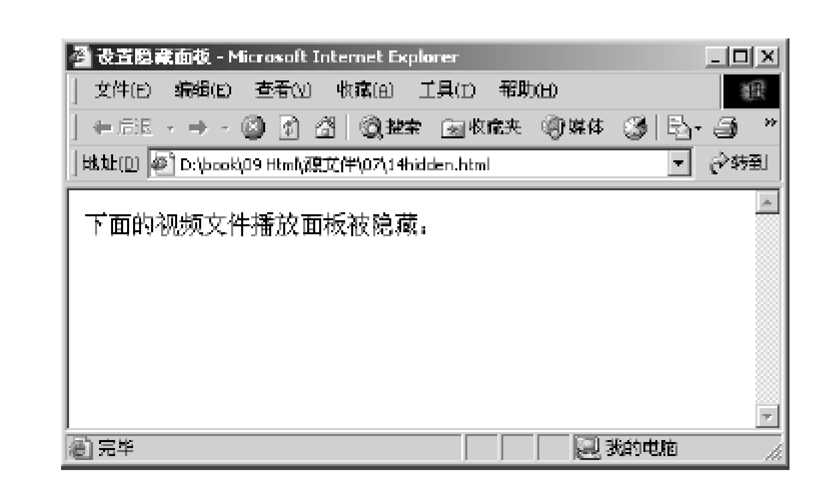


图 7-16　隐藏播放面板

7.3.5　添加其他类型的媒体文件

除了 avi 媒体文件之外，在网页中还可以嵌入 flash、mpeg 等类型的媒体文件，方法与 avi 媒体文件相同。

实例代码：

```
<html>
<head>
<title>嵌入多媒体文件</title>
</head>
<body>
    下面嵌入不同类型的媒体文件：<br>
    <embed src="exam01.swf" width=300></embed>
    <embed src="exam01.mpg" width=300></embed>
</body>
</html>
```

运行这段代码，看到在页面中添加了两种不同类型的媒体文件，如图 7-17 所示。其中，左侧的媒体文件为 Flash 类型，右侧的文件为 mpeg 类型。

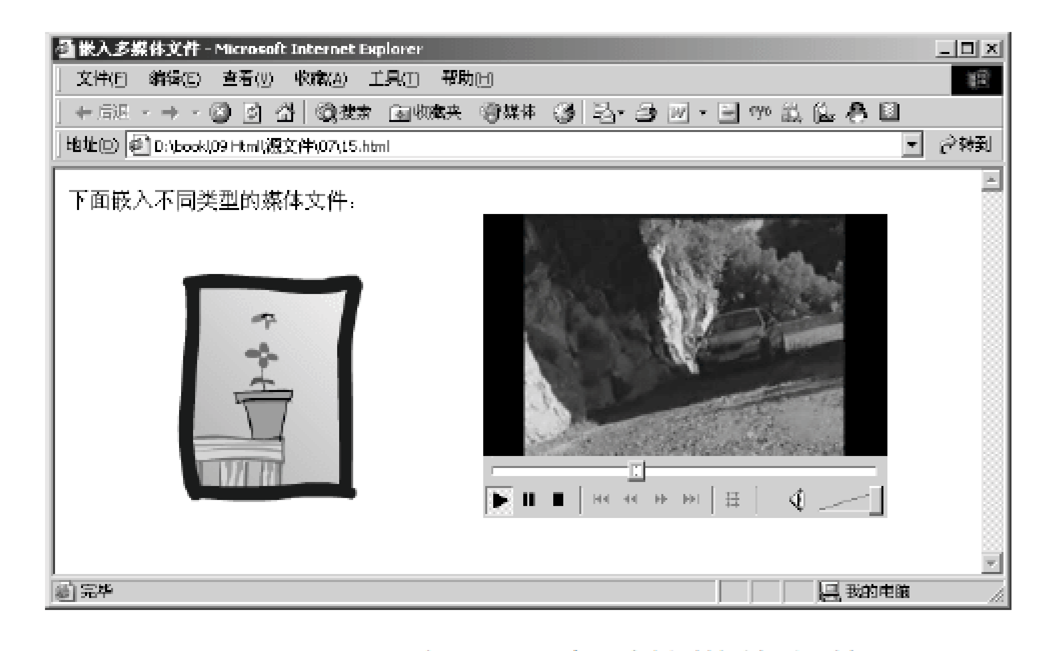


图 7-17　添加不同类型的媒体文件

第 8 章

表格的应用

- 建立表格
- 设置表格基本属性
- 表格的边框
- 设定表格背景
- 修改表格的行属性
- 调整单元格属性
- 表格的结构
- 表格的嵌套
- 层标记——div

表格是 HTML 的一项非常重要的功能，利用其多种属性能够设计出多样化的表格，可以说表格是网页排版的灵魂。同时由于表格包含的功能比较多，读者需要仔细学习才能掌握。

8.1　建 立 表 格

8.1.1　添加表格——table、tr、td

表格常用来对页面进行排版，在表格中一般通过 3 个标记来构建，分别是表格标记、行标记和单元格标记。其中表格标记是<table>和</table>，表格的其他各种属性都要在表格的开始标记<table>和表格的结束标记</table>之间才有效。下面首先介绍如何创建表格。

语法：

```
<table>
    <tr>
        <td>单元格内的文字</td>
        <td>单元格内的文字</td>
        ……
    </tr>
    <tr>
        <td>单元格内的文字</td>
        <td>单元格内的文字</td>
        ……
    </tr>
    ……
</table>
```

说明：<table>标记和</table>标记分别标志着一个表格的开始和结束；而<tr>和</tr>则分别表示表格中一行的开始和结束，在表格中包含几组<tr>…</tr>，就表示该表格为几行；<td>和</td>表示一个单元格的起始和结束，也可以说表示一行中包含了几列。

实例代码：

```
<html>
<head>
<title>添加表格</title>
</head>
<body>
    <h3>下面插入了一个表格：</h3>
    <table>
        <tr>
            <td>这是表格中的第一个单元格</td>
            <td>第一行中的第二个单元格</td>
        </tr>
        <tr>
            <td>这是表格的第二行</td>
            <td>第二行中的第二个单元格</td>
```

```
        </tr>
    </table>
</body>
</html>
```

运行这段代码，可以看到在网页中添加了一个两行两列的表格，但是这个表格没有边框线，如图 8-1 所示。

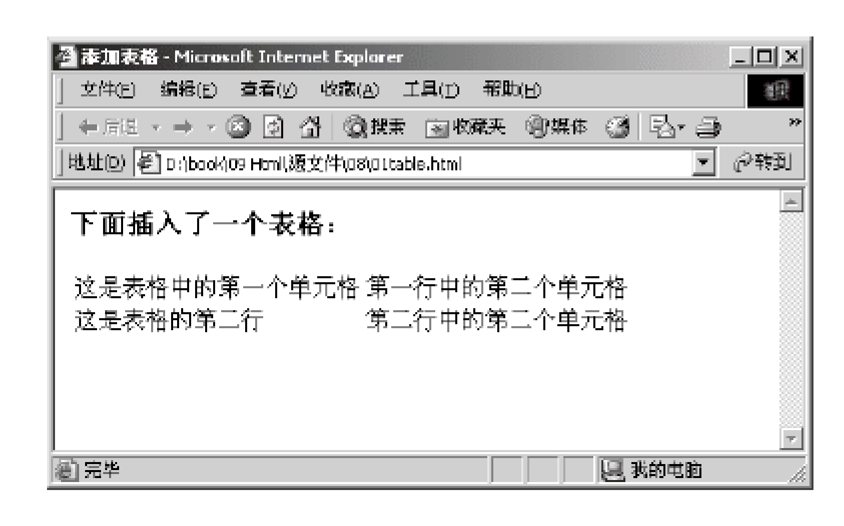


图 8-1　添加表格

8.1.2　设置表格的标题——caption

表格中除了<td>和</td>可用来设置表格的单元格外，还可以通过 caption 来设置特殊的一种单元格——标题单元格。表格的标题一般位于整个表格的第一行，为表格标示一个标题行，如同在表格上方加一个没有边框的行，通常用来存放表格标题。

语法：<caption>表格的标题</caption>

下面给本章第一个实例添加一个标题。

实例代码：

```
<html>
<head>
<title>添加表格标题</title>
</head>
<body>
    <h3>下面插入了一个表格：</h3>
    <table>
        <caption>添加表格的实例</caption>
        <tr>
            <td>这是表格中的第一个单元格</td>
            <td>第一行中的第二个单元格</td>
        </tr>
        <tr>
            <td>这是表格的第二行</td>
            <td>第二行中的第二个单元格</td>
        </tr>
    </table>
</body>
</html>
```

运行这段代码，看到在表格内容的上方一行添加了一个标题“添加表格的实例”，这一行标题默

认情况下居中显示，如图 8-2 所示。

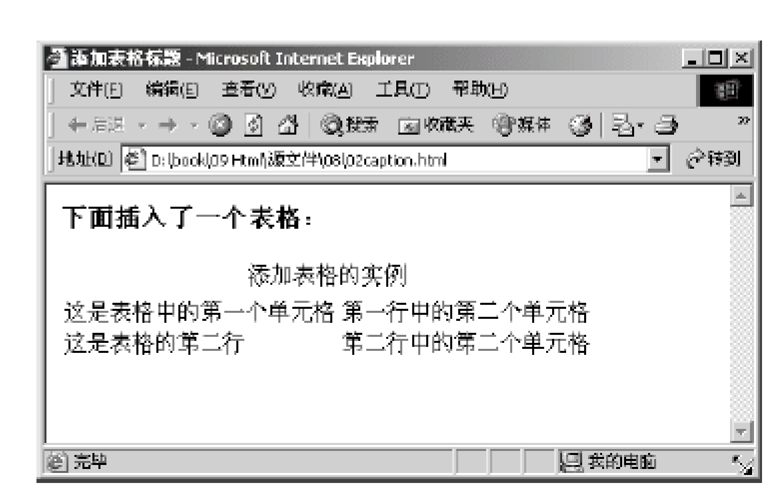


图 8-2　添加表格的标题

8.1.3　表格的表头——th

在表格中还有一种特殊的单元格，称为表头。表格的表头一般位于第一行和第一列，用来表明这一行的内容类别，用<th>和</th>标记来表示。表格的表头与<td>标记使用方法相同，但是表头的内容是加粗显示的。

语法：

```
<table>
    <tr>
        <th>表格的表头</th>
        <th>表格的表头</th>
        ……
    </tr>
    <tr>
        <td>单元格内的文字</td>
        <td>单元格内的文字</td>
        ……
    </tr>
    ……
</table>
```

实例代码：

```
<html>
<head>
<title>添加表格表头</title>
</head>
<body>
    <h3>下面公布了某中学期中考试的成绩：</h3>
    <table>
        <caption>期中考试成绩表</caption>
        <tr>
            <th>姓名</th>
            <th>语文</th>
```

```
            <th>数学</th>
            <th>英语</th>
            <th>物理</th>
            <th>化学</th>
        </tr>
        <tr>
            <td>章弹来</td>
            <td>94</td>
            <td>79</td>
            <td>93</td>
            <td>72</td>
            <td>75</td>
        </tr>
        <tr>
            <td>冯童</td>
            <td>79</td>
            <td>85</td>
            <td>74</td>
            <td>59</td>
            <td>73</td>
        </tr>
        <tr>
            <td>李四国</td>
            <td>94</td>
            <td>99</td>
            <td>95</td>
            <td>89</td>
            <td>83</td>
        </tr>
    </table>
</body>
</html>
```

运行程序，看到在表格中包含一行加粗字体，这就是表格的表头，如图 8-3 所示。

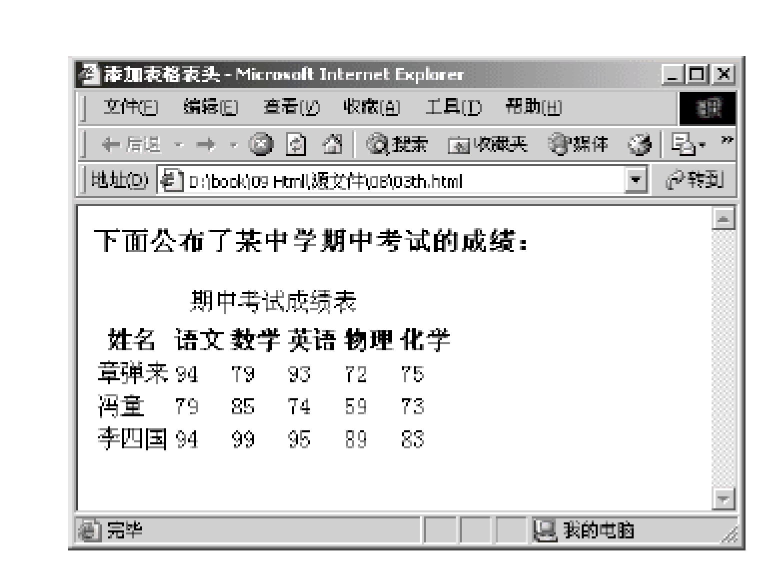


图 8-3　表格的表头

8.2　设置表格基本属性

表格的基本属性包括表格的大小和对齐方式，下面一一加以说明。

8.2.1　设置表格宽度——width

默认情况下，表格的宽度是与表格内的文字相关的，是根据内容自动调整的。如果想要指定表格的宽度，可以为表格添加 width 参数。

语法：<table width=表格宽度>

说明：表格宽度的值可以是具体的像素数，也可以设置为浏览器的百分比数。

实例代码：

```
<html>
<head>
<title>设置表格的宽度</title>
</head>
<body>
	<!--设置表格的宽度为浏览器的 90%-->
	<table width=90%>
		<caption>期中考试成绩表</caption>
		<tr>
			<th>姓名</th>
			<th>语文</th>
			<th>数学</th>
			<th>英语</th>
			<th>物理</th>
			<th>化学</th>
		</tr>
		<tr>
			<td>章弹来</td>
			<td>94</td>
			<td>79</td>
			<td>93</td>
			<td>72</td>
			<td>75</td>
		</tr>
		<tr>
			<td>冯童</td>
			<td>79</td>
			<td>85</td>
			<td>74</td>
			<td>59</td>
			<td>73</td>
		</tr>
		<tr>
			<td>李四国</td>
			<td>94</td>
			<td>99</td>
			<td>95</td>
			<td>89</td>
			<td>83</td>
		</tr>
	</table>
</body>
</html>
```

运行这段代码，看到表格的效果如图 8-4 所示。调整浏览器的宽度后，表格的宽度也会随之变化，如图 8-5 所示。

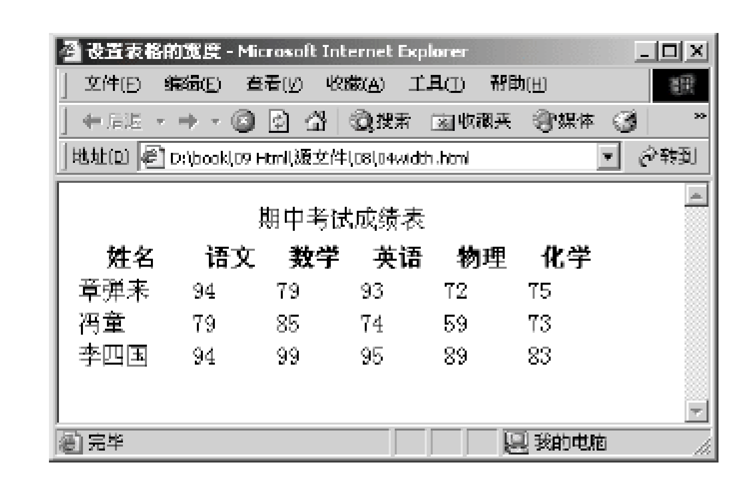



图 8-4　运行代码后的表格效果

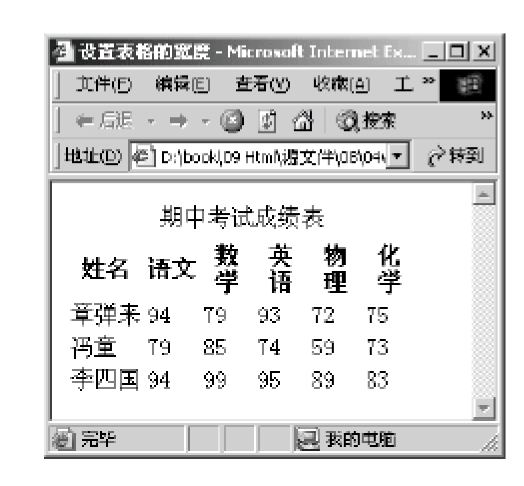


图 8-5　调整浏览器宽度后的表格效果

如果将表格中的宽度值设置为固定的像素数，那么当浏览器大小变化时，表格不会随之变化。

8.2.2　设置表格高度——height

设置表格高度的方法与设置表格宽度的方法相同，也可以将表格的高度设置为浏览器高度的百分比或者是固定的像素数。

语法：<table height=表格高度>

实例代码：

```
<html>
<head>
<title>设置表格的高度</title>
</head>
<body>
	<!--设置表格的宽度为浏览器的 90%，高度为 200 像素-->
	<table width=90% height=200>
		<caption>期中考试成绩表</caption>
		<tr>
			<th>姓名</th>
			<th>语文</th>
			<th>数学</th>
			<th>英语</th>
			<th>物理</th>
			<th>化学</th>
		</tr>
		<tr>
			<td>章弹来</td>
			<td>94</td>
			<td>79</td>
			<td>93</td>
			<td>72</td>
			<td>75</td>
		</tr>
		<tr>
			<td>冯童</td>
			<td>79</td>
			<td>85</td>
			<td>74</td>
```

```
                <td>59</td>
                <td>73</td>
            </tr>
            <tr>
                <td>李四国</td>
                <td>94</td>
                <td>99</td>
                <td>95</td>
                <td>89</td>
                <td>83</td>
            </tr>
        </table>
</body>
</html>
```

运行这段代码，可以看到由于将表格高度设为了固定的像素数，无论浏览器如何变化，表格的高度都保持不变，如图 8-6 所示。

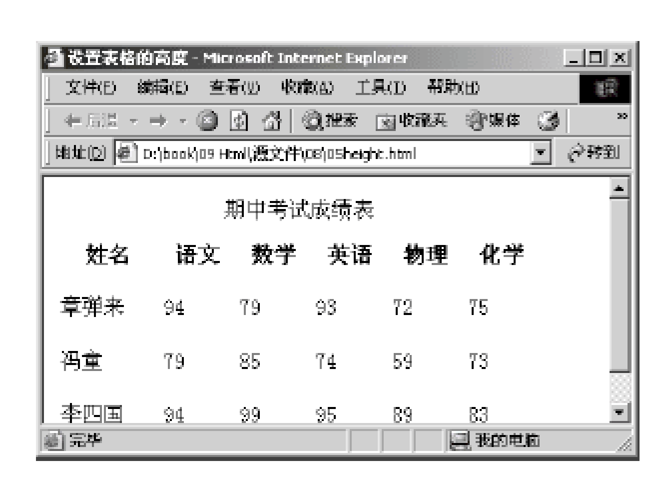


图 8-6　设置表格的高度

8.2.3　表格对齐方式——align

表格的对齐方式用于设置整个表格在网页中的位置。

语法：<table align="表格对齐方式">

说明：align 参数可以取值为 left、center 或者 right。

实例代码：

```
<html>
<head>
<title>设置表格对齐方式</title>
</head>
<body>
        <table align="center" width=600>
            <caption>通讯录</caption>
            <tr>
                <th>姓名</th>
                <th>地址</th>
                <th>电话</th>
                <th>电子邮件</th>
            </tr>
            <tr>
                <td>丁子恒</td>
                <td>北京市海淀区东王庄 183 号</td>
                <td>010-83546675</td>
                <td>ddzz@yahoo.com.cn</td>
            </tr>
            <tr>
                <td>冯岗</td>
```

```
                <td>北京市景山公园 3 号楼 1106 室</td>
                <td>010-63546874</td>
                <td>fengg165@jingshan.net</td>
            </tr>
            <tr>
                <td>王奇安</td>
                <td>南京市雨花路 875 号</td>
                <td>0352-87457454</td>
                <td>wqa@163.com</td>
            </tr>
        </table>
    </body>
    </html>
```

运行这段代码，可以看到表格的内容居中显示，如图 8-7 所示。



图 8-7　设置表格的对齐方式

8.3　表格的边框

表格除了设置基本属性外，还可以设置边框效果，包括颜色、宽度等。

8.3.1　设置表格的边框宽度——border

默认情况下，表格是不显示边框的。为了使表格更加清晰，可以使用 border 参数设置边框的宽度。

语法：<table border=边框宽度>

说明：只有设定了 border 参数，且其值不为 0，在网页中才能显示出表格的边框。border 的单位为像素。

实例代码：

```
<html>
<head>
<title>设置表格边框</title>
</head>
<body>
    <table width=600 border=1>
        <caption>通讯录</caption>
        <tr>
            <th>姓名</th>
            <th>地址</th>
            <th>电话</th>
            <th>电子邮件</th>
        </tr>
        <tr>
            <td>丁子恒</td>
            <td>北京市海淀区东王庄 183 号</td>
```

```
            <td>010-83546675</td>
            <td>ddzz@yahoo.com.cn</td>
        </tr>
        <tr>
            <td>冯岗</td>
            <td>北京市景山公园 3 号楼 1106 室</td>
            <td>010-63546874</td>
            <td>fengg165@jingshan.net</td>
        </tr>
        <tr>
            <td>王奇安</td>
            <td>南京市雨花路 875 号</td>
            <td>0352-87457454</td>
            <td>wqa@163.com</td>
        </tr>
    </table>
</body>
</html>
```

运行程序，表格的边框宽度为 1 像素，效果如图 8-8 所示。由于第一行“通讯录”为表格的标题，因此其周围并没有边框。



图 8-8　设置表格的边框

8.3.2　表格边框颜色——bordercolor

默认情况下，边框的颜色是灰色的，为了让表格更鲜明，可以使用 bordercolor 参数设置不同的表格边框颜色。但是设置边框颜色的前提是边框宽度不能为 0，否则无法显示出应有的效果。

语法：<table border=边框宽度 bordercolor= "边框颜色" >

说明：在该语法中，边框宽度不能为 0，边框颜色为 16 位颜色代码。

实例代码：

```
<html>
<head>
<title>设置边框颜色</title>
</head>
<body>
    <table width=600 border=3 bordercolor="#9933CC">
        <caption>A 厂某月工资表</caption>
        <tr>
            <th>姓名</th>
```

```
            <th>基本工资</th>
            <th>岗位工资</th>
            <th>本月奖金</th>
        </tr>
        <tr>
            <td>崔维军</td>
            <td>1000</td>
            <td>1500</td>
            <td>1300</td>
        </tr>
        <tr>
            <td>范勇</td>
            <td>800</td>
            <td>1000</td>
            <td>850</td>
        </tr>
        <tr>
            <td>王奇安</td>
            <td>800</td>
            <td>1200</td>
            <td>1150</td>
        </tr>
    </table>
</body>
</html>
```

运行这段代码，看到表格的边框颜色发生了变化，如图 8-9 所示。

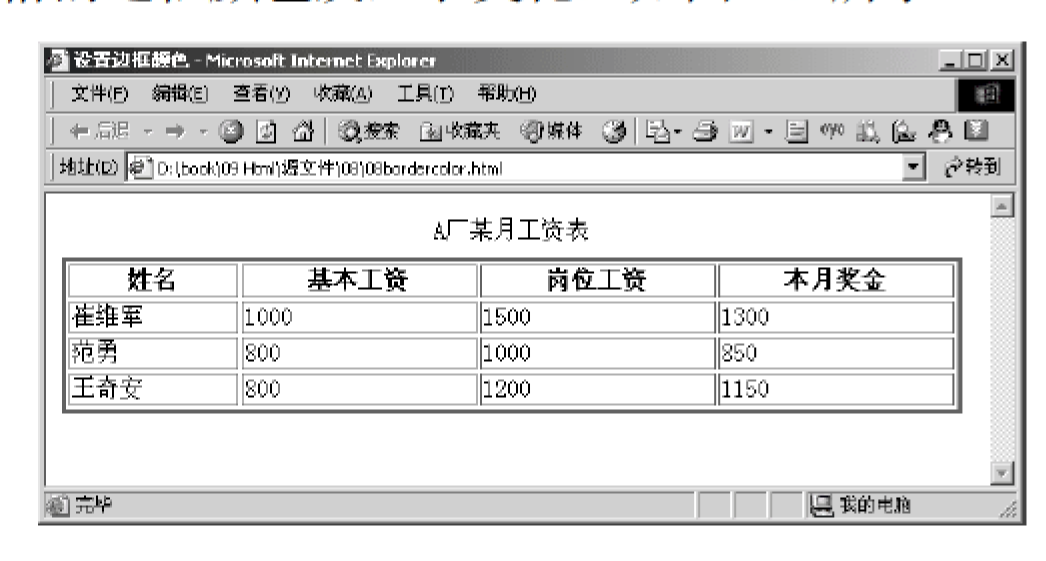


图 8-9　设置表格边框颜色

8.3.3　内框宽度——cellspacing

表格的内框宽度是指表格内部各个单元格之间的宽度。

语法：<table cellspacing=内框宽度>

说明：内框宽度的单位为像素。

实例代码：

```
<html>
<head>
<title>设置表格内框宽度</title>
```

```
</head>
<body>
    <table width=660 border=1 bordercolor="#990000" cellspacing=10>
        <caption>通讯录</caption>
        <tr>
            <th>姓名</th>
            <th>地址</th>
            <th>电话</th>
            <th>电子邮件</th>
        </tr>
        <tr>
            <td>丁子恒</td>
            <td>北京市海淀区东王庄 183 号</td>
            <td>010-83546675</td>
            <td>ddzz@yahoo.com.cn</td>
        </tr>
        <tr>
            <td>冯岗</td>
            <td>北京市景山公园 3 号楼 1106 室</td>
            <td>010-63546874</td>
            <td>fengg165@jingshan.net</td>
        </tr>
        <tr>
            <td>王奇安</td>
            <td>南京市雨花路 875 号</td>
            <td>0352-87457454</td>
            <td>wqa@163.com</td>
        </tr>
    </table>
</body>
</html>
```

运行这段代码，可以看到表格中单元格之间的距离拉大了，如图 8-10 所示。

如果将表格的内框宽度更改为 0，效果如图 8-11 所示。



图 8-10　设置内框宽度



图 8-11　调整表格的内框宽度

8.3.4　表格内文字与边框距离——cellpadding

在默认情况下，表格内的文字会紧贴着表格的边框，这样看上去非常拥挤。可以使用 cellpadding

参数来调整这一距离。

语法：<table cellpadding=文字与边框的距离值>

说明：文字与边框的距离以像素为单位，一般可以根据需要设置，但要注意不能过大。因为这个值不只对左右距离有效，同时也设置了上下边框与文字的间隔。

实例代码：

```
<html>
<head>
<title>设置文字与边框的距离</title>
</head>
<body>
    <table border=1 bordercolor="#990000" cellspacing=3 cellpadding=10>
        <caption>通讯录</caption>
        <tr>
            <th>姓名</th>
            <th>地址</th>
            <th>电子邮件</th>
        </tr>
        <tr>
            <td>丁子恒</td>
            <td>北京市海淀区东王庄 183 号</td>
            <td>ddzz@yahoo.com.cn</td>
        </tr>
        <tr>
            <td>冯岗</td>
            <td>北京市景山公园 3 号楼 1106 室</td>
            <td>fengg165@jingshan.net</td>
        </tr>
        <tr>
            <td>王奇安</td>
            <td>南京市雨花路 875 号</td>
            <td>wqa@163.com</td>
        </tr>
    </table>
</body>
</html>
```

运行这段代码，效果如图 8-12 所示。

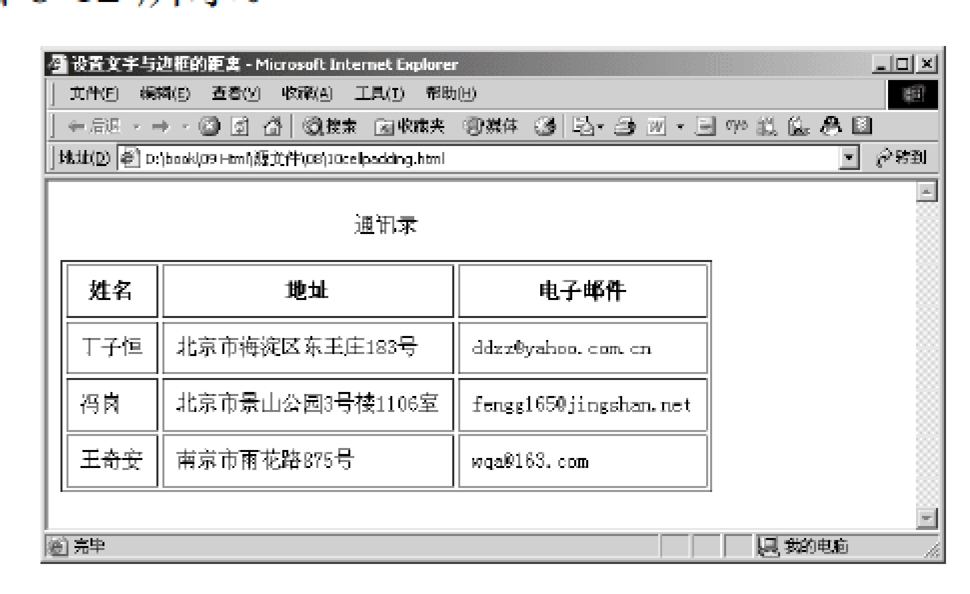


图 8-12　设置文字与边框的距离

8.4　设定表格背景

为了突出显示表格，还可以为表格设置与页面不同的背景。

8.4.1　设置表格背景色——bgcolor

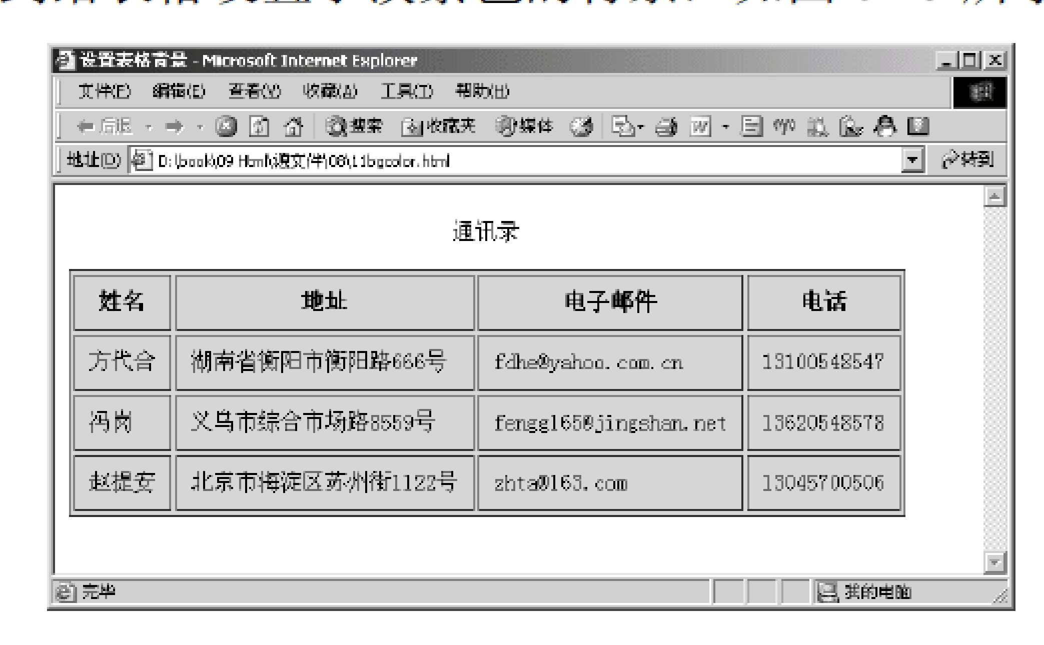
设置表格背景，最简单的就是给表格设置背景颜色。

语法：<table bgcolor="颜色代码">

实例代码：

```
<html>
<head>
<title>设置表格背景</title>
</head>
<body>
	<table bgcolor="#DDCCFF" border=1 bordercolor="#990000" cellspacing=3 cellpadding=10>
		<caption>通讯录</caption>
		<tr>
			<th>姓名</th>
			<th>地址</th>
			<th>电子邮件</th>
			<th>电话</th>
		</tr>
		<tr>
			<td>方代合</td>
			<td>湖南省衡阳市衡阳路 666 号</td>
			<td>fdhe@yahoo.com.cn</td>
			<td>13100548547</td>
		</tr>
		<tr>
			<td>冯岗</td>
			<td>义乌市综合市场路 8559 号</td>
			<td>fengg165@jingshan.net</td>
			<td>13620548578</td>
		</tr>
		<tr>
			<td>赵提安</td>
			<td>北京市海淀区苏州街 1122 号</td>
			<td>zhta@163.com</td>
			<td>13045700506</td>
		</tr>
	</table>
</body>
</html>
```

运行这段代码，可以看到给表格设置了淡紫色的背景，如图 8-13 所示。



姓名	地址	电子邮件	电话
方代合	湖南省衡阳市衡阳路666号	fdhe@yahoo.com.cn	13100548547
冯岗	义乌市综合市场路8559号	fengg165@jingshan.net	13620548578
赵提安	北京市海淀区苏州街1122号	zhta@163.com	13045700506

图 8-13　设置表格背景

8.4.2　设置表格的背景图像——background

除了可以为表格设置背景色之外，还可以设置一个背景图像，让表格更加绚丽。

语法：<table background="背景图片的地址">

说明：背景图片的地址可以设置为相对地址，也可以设置为绝对地址。

实例代码：

```
<html>
<head>
<title>设置表格背景</title>
</head>
<body>
    <table background="pic01.jpg" border=1 bordercolor="#660000" cellpadding=5>
        <caption>通讯录</caption>
        <tr>
            <th>姓名</th>
            <th>地址</th>
            <th>电子邮件</th>
        </tr>
        <tr>
            <td>方代合</td>
            <td>湖南省衡阳市衡阳路 666 号</td>
            <td>fdhe@yahoo.com</td>
        </tr>
        <tr>
            <td>冯岗</td>
            <td>义乌市综合市场路 8559 号</td>
            <td>fengg@jingshan.net</td>
        </tr>
        <tr>
            <td>赵提安</td>
            <td>北京市海淀区苏州街 1122 号</td>
            <td>zhta@163.com</td>
        </tr>
```

```
    </table>
</body>
</html>
```

运行这段代码，可以看到表格的背景图像如图 8-14 所示。



图 8-14　设置表格的背景图像

8.5　修改表格的行属性

设定了表格的整体属性后，还可以对单独的一行表格进行属性设置。

8.5.1　高度的控制——height

在网页中常常遇到一些表格中某一行高度和其他行不同的情况，这时就需要使用 height 参数。

语法：<tr height=表格中某行高度>

说明：这一参数只对设置的这一行有效。

实例代码：

```
<html>
<head>
<title>设置行高度</title>
</head>
<body>
    <table border=1 bordercolor="#660000" cellpadding=5>
        <caption>我国著名词人——李清照</caption>
        <tr height=90>
            <td>朝代</td>
            <td>宋代</td>
        </tr>
        <tr>
            <td>著作</td>
            <td>如梦令、永遇乐、点绛唇、蝶恋花、浣溪沙等</td>
        </tr>
        <tr height=160>
```

```
                <td>简介</td>
                <td> 李清照(1084-1155?)号易安居士，齐州章丘（今属山东济南）人，以词著称，有较高的艺术
造诣。<br>
                父李格非为当时著名学者，夫赵明诚为金石考据家。早期生活优裕，与明诚共同致力于书画金石
的搜集整理。金兵入据中原，流寓南方，明诚病死，境遇孤苦。所作词，前期多写其悠闲生活，后期多悲叹身世，
情调感伤，有的也流露出对中原的怀念。形式上善用白描手法，自辟途径，语言清丽。论词强调协律，崇尚典雅、
情致，提出词“别是一家”之说，反对以作诗文之法作词。</td>
            </tr>
        </table>
</body>
</html>
```

运行程序，效果如图 8-15 所示。在图中，第 1 行设置了 90 像素的高度；第 2 行没有设置高度值，是表格的默认高度；第 3 行设置了 160 像素的高度。

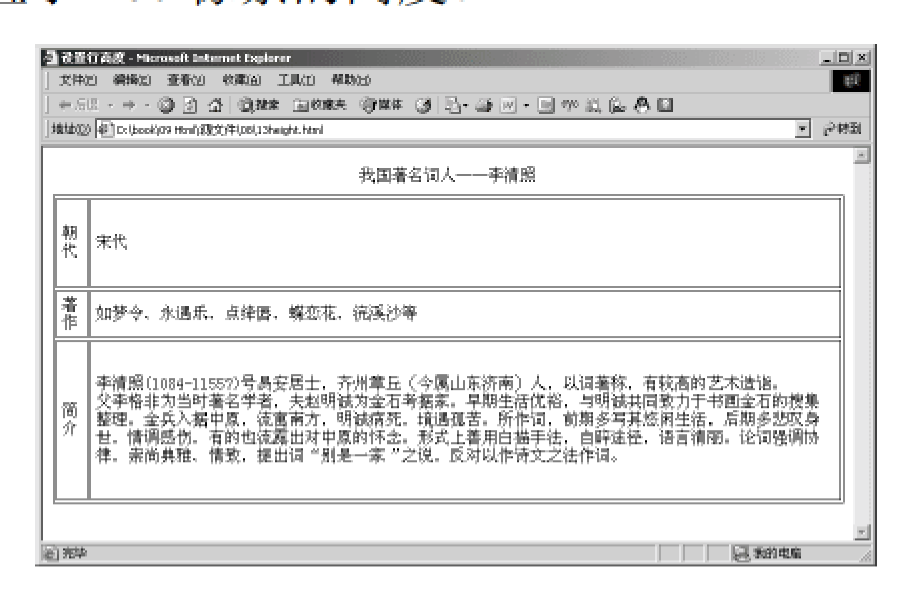


图 8-15　设置行高度

8.5.2　边框颜色——bordercolor

与表格相同，对表格的行来说，也可以单独设置其外框颜色。

语法：<tr bordercolor="颜色代码">

实例代码：

```
<html>
<head>
<title>设置行的边框颜色</title>
</head>
<body>
    <table width=500 border=1 bordercolor="00CCFF">
        <caption>A 厂某月工资表</caption>
        <tr>
            <th>姓名</th>
            <th>基本工资</th>
            <th>岗位工资</th>
            <th>本月奖金</th>
        </tr>
        <tr bordercolor="#990000">
            <td>崔维军</td>
            <td>1000</td>
```

```
                <td>1500</td>
                <td>1300</td>
            </tr>
            <tr>
                <td>范勇</td>
                <td>800</td>
                <td>1000</td>
                <td>850</td>
            </tr>
            <tr>
                <td>王奇安</td>
                <td>800</td>
                <td>1200</td>
                <td>1150</td>
            </tr>
        </table>
</body>
</html>
```

运行代码，效果如图 8-16 所示。其中第 2 行的表格边框颜色是单独设置的，与整个表格不同。这种方法常常用来突出显示表格中的某一行。



图 8-16　设置行的边框颜色

8.5.3　行背景——bgcolor、background

与设置行的边框颜色相同，行的背景色也可以单独设置。

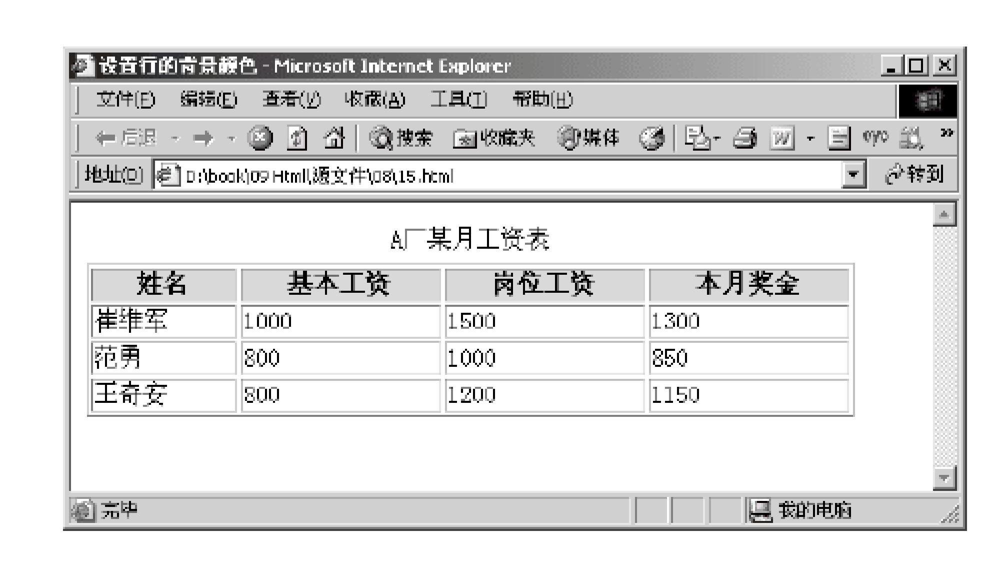
语法：<tr bgcolor="颜色代码">

实例代码：

```
<html>
<head>
<title>设置行的背景颜色</title>
</head>
<body>
        <table width=500 border=1>
            <caption>A 厂某月工资表</caption>
            <tr bgcolor="#FFEE00">
                <th>姓名</th>
```

```
            <th>基本工资</th>
            <th>岗位工资</th>
            <th>本月奖金</th>
        </tr>
        <tr>
            <td>崔维军</td>
            <td>1000</td>
            <td>1500</td>
            <td>1300</td>
        </tr>
        <tr>
            <td>范勇</td>
            <td>800</td>
            <td>1000</td>
            <td>850</td>
        </tr>
        <tr>
            <td>王奇安</td>
            <td>800</td>
            <td>1200</td>
            <td>1150</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，表格的第一行设置了单独的背景色，如图 8-17 所示。


A厂某月工资表

姓名	基本工资	岗位工资	本月奖金
崔维军	1000	1500	1300
范勇	800	1000	850
王奇安	800	1200	1150

图 8-17　设置行的背景色

8.5.4　行文字的水平对齐方式——align

表格中也可以为单独的一行设置特殊对齐方式。

语法：<tr align="水平对齐方式">

说明：这里水平对齐方式包含 3 种，分别为 left、center 和 right。

实例代码：

```
<html>
<head>
<title>设置行的水平对齐方式</title>
</head>
<body>
    <table border=1 bordercolor="#660000" cellpadding=5>
        <caption>我国著名词人——李清照</caption>
        <tr align="center">
            <td>朝代</td>
            <td>宋代</td>
        </tr>
        <tr align="right">
```

```
                <td>著作</td>
                <td>如梦令、永遇乐、点绛唇、蝶恋花、浣溪沙等</td>
            </tr>
            <tr align="left" height=90>
                <td>简介</td>
                <td> 李清照(1084-1155?)号易安居士，齐州章丘（今属山东济南）人，以词著称，有较高的艺术
造诣。</td>
            </tr>
        </table>
</body>
</html>
```

运行代码，效果如图 8-18 所示。其中第 1 行为水平居中；第 2 行为右对齐；第 3 行为左对齐。

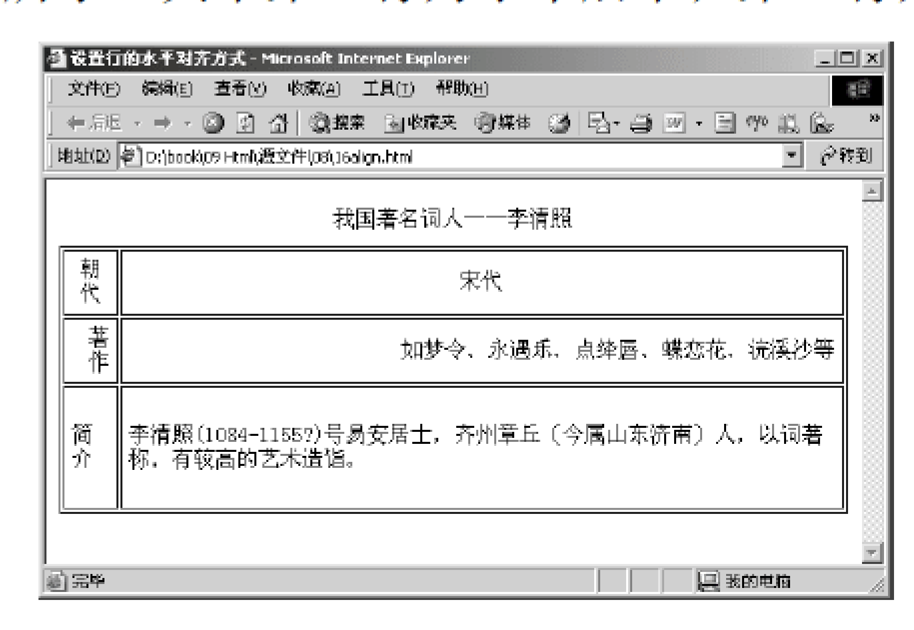


图 8-18　设置行的水平对齐方式

8.5.5　行文字的垂直对齐方式——valign

表格中也可以为单独的一行设置特殊对齐方式。

语法：<tr valign="垂直对齐方式">

说明：这里，垂直对齐方式可以取的值包含 3 种，分别为 top、middle 和 bottom。

实例代码：

```
<html>
<head>
<title>设置行的垂直对齐方式</title>
</head>
<body>
        <table border=1 bordercolor="#660000" cellpadding=5>
            <caption>我国著名词人——李清照</caption>
            <tr height=50 valign="top">
                <td>朝代</td>
                <td>宋代</td>
            </tr>
            <tr height=50 valign="center">
                <td>著作</td>
                <td>如梦令、永遇乐、点绛唇、蝶恋花、浣溪沙等</td>
            </tr>
            <tr height=90 valign="bottom">
```

```
            <td>简介</td>
            <td> 李清照(1084-1155?)号易安居士，齐州章丘（今属山东济南）人，以词著称，有较高的艺术造诣。</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，效果如图 8-19 所示。其中第 1 行为顶端对齐；第 2 行为垂直居中对齐；第 3 行为底端对齐。

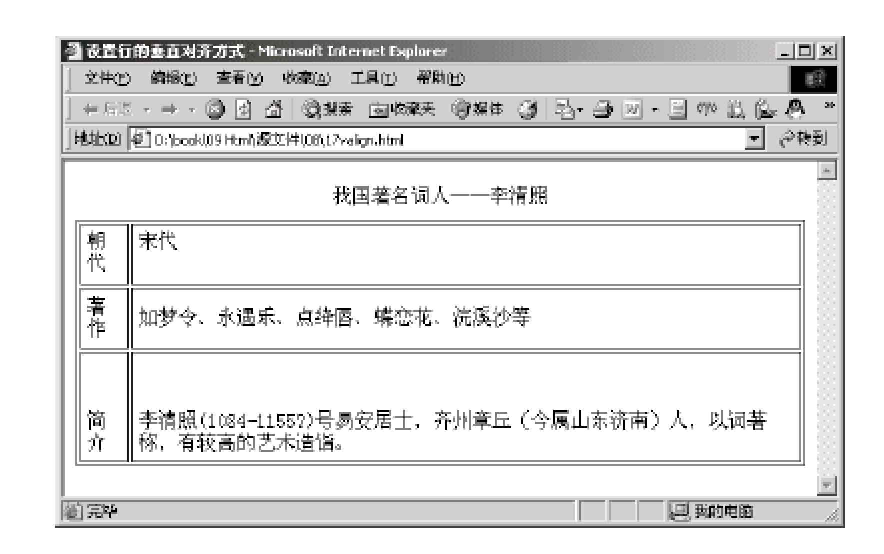


图 8-19　设置行文字的垂直对齐效果

8.5.6　设置表格标题的垂直对齐方式——valign

表格标题是一种特殊的行，这种行没有边框，但是依然可以设置对齐方式，包括设置水平对齐方式和垂直对齐方式。

由于表格标题也是行的一种，其水平对齐方式的设置与其他行相同，通过 align 参数来设置，这里不再重复说明。此处讲解以下标题的垂直对齐方式，虽然同样使用 valign 参数来设置，但与其他行不同的是，标题的垂直对齐是指标题位于表格的上方或下方。

语法：<caption valign="垂直对齐方式">表格的标题</caption>

说明：这里垂直对齐方式的值可以是 top 或者 bottom。取值为 top，是将标题文字设置在表格的上方；取值为 bottom，是将标题文字设置在表格的下方。

实例代码：

```
<head>
<title>设置表格标题的垂直对齐方式</title>
</head>
<body>
    <table border=1 bordercolor="#660000" cellspacing=0 cellpadding=5>
        <caption valign="bottom">某电脑公司销售清单</caption>
        <tr>
            <td>销售对象</td>
            <td>类别</td>
            <td>产品型号</td>
            <td>数量</td>
            <td>单价</td>
        </tr>
```

```
        <tr>
            <td>单位</td>
            <td>台式电脑</td>
            <td>G280</td>
            <td>5</td>
            <td>6788</td>
        </tr>
        <tr>
            <td>个人</td>
            <td>电脑配件</td>
            <td>GT2805</td>
            <td>1</td>
            <td>530</td>
        </tr>
        <tr>
            <td>个人</td>
            <td>笔记本电脑</td>
            <td>TN3765</td>
            <td>1</td>
            <td>18999</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，效果如图 8-20 所示。

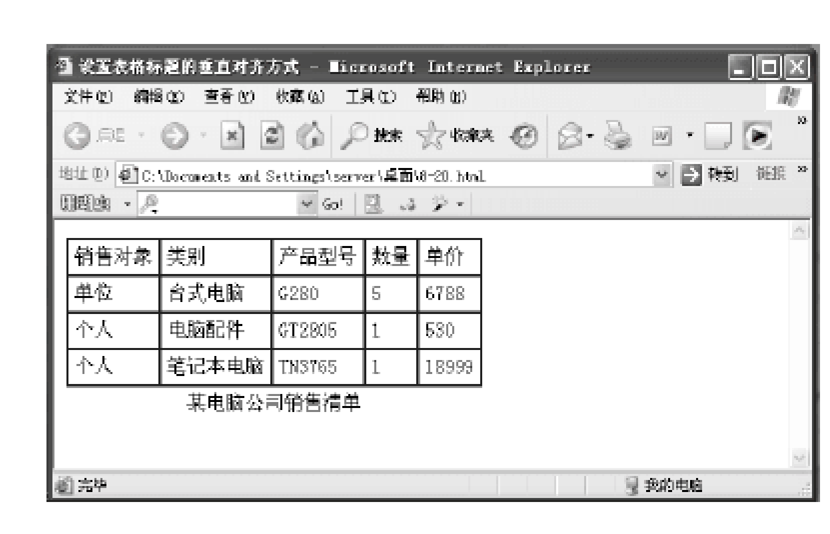


图 8-20　设置表格标题的垂直对齐方式

8.6　调整单元格属性

表格中另外一种元素就是单元格，单元格的属性标记和行标记非常相似。

8.6.1　单元格大小——width、height

默认情况下，单元格的大小会根据内容自动调整，也可以进行手动调整。

语法：<td width=单元格宽度 height=单元格高度>

说明：单元格高度和宽度的单位是像素，而对一个单元格的设置往往会影响多个单元格。例如设置了第 1 行的第 1 个单元格的宽度，其他行的第 1 个单元格宽度往往也会随之变化。

实例代码：

```
<head>
<title>设置单元格大小</title>
</head>
<body>
    <table border=1 bordercolor="#660000" cellspacing=0 cellpadding=5>
```

```
        <caption>某电脑公司销售清单</caption>
        <tr bgcolor="#DDCCFF">
            <td width=90 height=20>销售对象</td>
            <td width=110>类别</td>
            <td>产品型号</td>
            <td>数量</td>
            <td>单价</td>
        </tr>
        <tr>
            <td height=40>单位</td>
            <td>台式电脑</td>
            <td width=130>G280</td>
            <td width=70>5</td>
            <td>6788</td>
        </tr>
        <tr>
            <td>个人</td>
            <td>电脑配件</td>
            <td>GT2805</td>
            <td>1</td>
            <td height=70 width=120>530</td>
        </tr>
        <tr>
            <td>个人</td>
            <td height=35>笔记本电脑</td>
            <td>TN3765</td>
            <td>1</td>
            <td>18999</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，效果如图 8-21 所示。

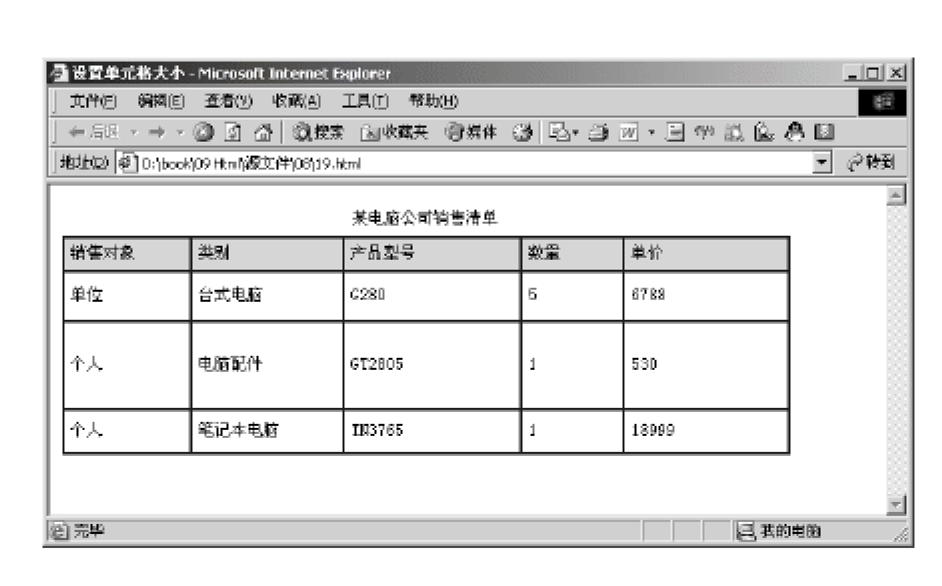


图 8-21　设置单元格大小

8.6.2　水平跨度——colspan

单元格水平跨度是指在复杂的表格结构中，有些单元格是跨多个列的。

语法：<td colspan=跨的列数>

说明：在这里，跨的列数就是这个单元格所跨列的个数，也可以说是单元格向右打通的单元格个数。

实例代码：

```
<head>
<title>设置单元格列跨度</title>
</head>
<body>
    <table border=1 bordercolor="#660000" cellspacing=0 cellpadding=5>
```

```
        <caption>某电脑公司销售清单</caption>
        <tr bgcolor="#CCAAFF">
            <td width=80></td>
            <td colspan=3 width=400 align="center">第一季度</td>
        </tr>
        <tr bgcolor="#DDCCFF">
            <td>产品型号</td>
            <td>1 月份</td>
            <td>2 月份</td>
            <td>3 月份</td>
        </tr>
        <tr>
            <td>G280</td>
            <td>1500</td>
            <td>1800</td>
            <td>2350</td>
        </tr>
        <tr>
            <td>GT2805</td>
            <td>985</td>
            <td>908</td>
            <td>1200</td>
        </tr>
        <tr>
            <td>TN3765</td>
            <td>700</td>
            <td>750</td>
            <td>600</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，效果如图 8-22 所示，其中第 1 行的第 2 个单元格跨了 3 列。

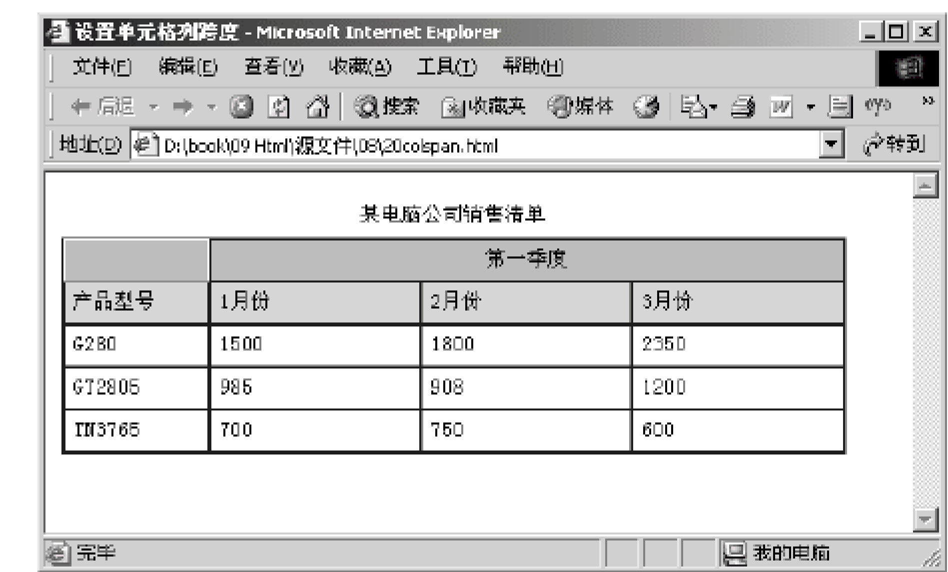


图 8-22　设置单元格列跨度

8.6.3　垂直跨度——rowspan

单元格除了可以在水平方向上跨列，还可在垂直方向上跨行。跨行设置需要使用 rowspan 参数。

语法：<td rowspan=单元格跨行数>

说明：与水平跨度相对应，rowspan 设置的是单元格在垂直方向上跨行的个数，也可以说是单元格向下打通的单元格个数。

实例代码：

```
<head>
<title>设置单元格行跨度</title>
</head>
```

```
<body>
    <table border=1 bordercolor="#660000" cellspacing=0 cellpadding=5>
        <caption>某网上书店销售分类</caption>
        <tr bgcolor="#DDDDFF">
            <td width=130>类别</td>
            <td width=290>子类别</td>
        </tr>
        <tr>
            <td rowspan=3>电脑书籍</td>
            <td>编程类</td>
        </tr>
        <tr>
            <td>图形图像类</td>
        </tr>
        <tr>
            <td>数据库类</td>
        </tr>
        <tr>
            <td rowspan=2>考试专区</td>
            <td>中考高考</td>
        </tr>
        <tr>
            <td>考研类  </td>
        </tr>
    </table>
</body>
</html>
```

运行代码，效果如图 8-23 所示。

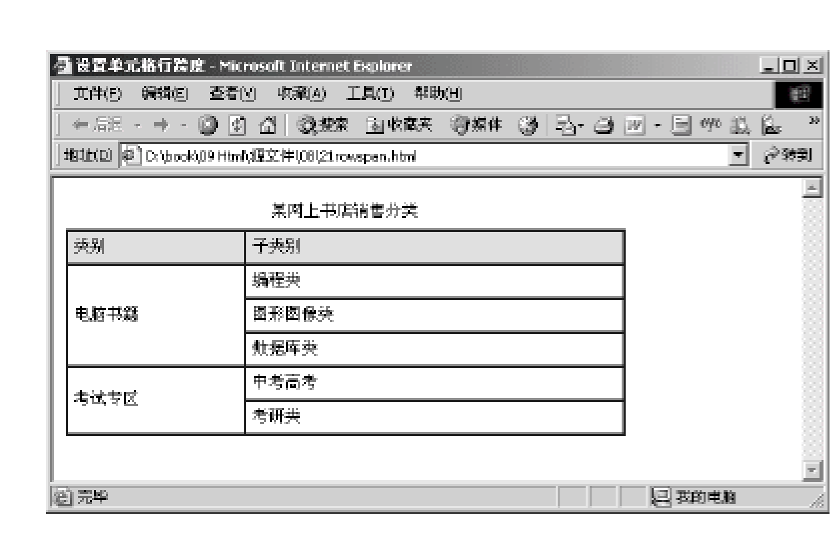


图 8-23　设置单元格的行跨度

8.6.4　对齐方式——align、valign

单元格的对齐方式设置与行的对齐方式设置方法相似。

语法：<td align="水平对齐方式" valign="垂直对齐方式">

说明：在该语法中，水平对齐方式的取值可以是 left、center 或 right；垂直对齐方式的取值可以是 top、middle 或 bottom。

实例代码：

```
<head>
<title>设置单元格的文字对齐方式</title>
</head>
<body>
    <table border=1 bordercolor="#660000" cellspacing=0 cellpadding=5 height=200 width=550>
        <caption>某电脑公司销售清单</caption>
        <tr bgcolor="#DDCCFF" >
            <td>销售对象</td>
            <td >类别</td>
            <td>产品型号</td>
```

```
            <td>数量</td>
            <td>单价</td>
        </tr>
        <tr>
            <td>单位</td>
            <td>台式电脑</td>
            <td>G280</td>
            <td>5</td>
            <td>6788</td>
        </tr>
        <tr>
            <td>个人</td>
            <td>电脑配件</td>
            <td>GT2805</td>
            <td>1</td>
            <td>530</td>
        </tr>
        <tr>
            <td>个人</td>
            <td>笔记本电脑</td>
            <td>TN3765</td>
            <td>1</td>
            <td>18999</td>
        </tr>
    </table>
</body>
</html>
```

运行程序，效果如图 8-24 所示。图中对不同单元格的对齐方式进行了不同的设置，效果也不相同。

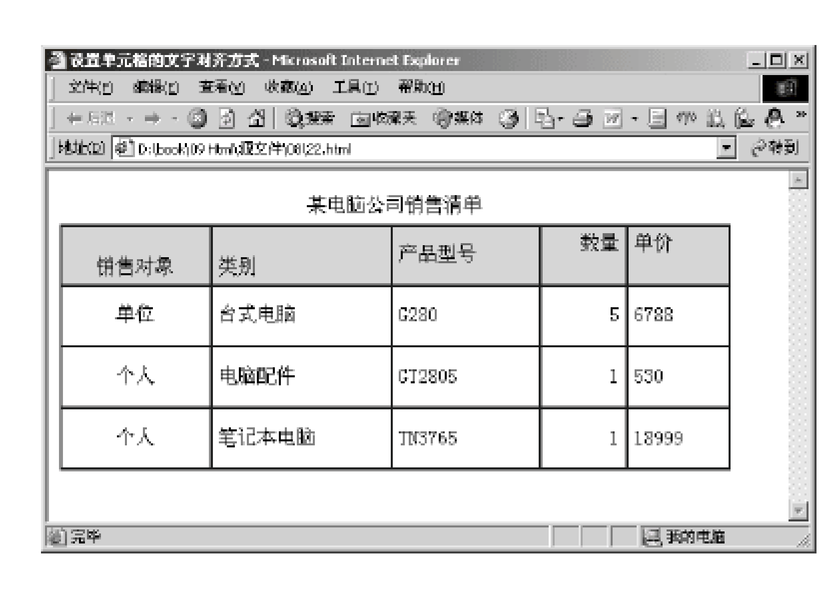


图 8-24　设置单元格对齐方式

8.6.5　设置单元格的背景色

为了增加表格的绚丽，可以为不同的单元格分别设置不同的背景颜色。

语法：<td bgcolor="颜色代码">

实例代码：

```
<head>
<title>设置单元格的背景色</title>
</head>
<body>
    <table border=1 bordercolor="#660000" cellspacing=0 cellpadding=5 height=200 width=550>
        <caption>某电脑公司销售清单</caption>
        <tr bgcolor="#FFF2C1" >
            <td>销售对象</td>
            <td>类别</td>
            <td>产品型号</td>
            <td>数量</td>
```

```
            <td>单价</td>
        </tr>
        <tr>
            <td bgcolor="#71E2FF">单位</td>
            <td bgcolor="#B382EE">台式电脑</td>
            <td bgcolor="#7EF1C0">G280</td>
            <td bgcolor="#78DC87">5</td>
            <td bgcolor="#F19B7E">6788</td>
        </tr>
        <tr>
            <td bgcolor="#7EF1CF">个人</td>
            <td bgcolor="#E49B7E">电脑配件</td>
            <td bgcolor="#EF81C6">GT2805</td>
            <td bgcolor="#B6B0E3">1</td>
            <td bgcolor="#B182FF">530</td>
        </tr>
    </table>
</body>
</html>
```

运行这段代码，效果如图 8-25 所示。

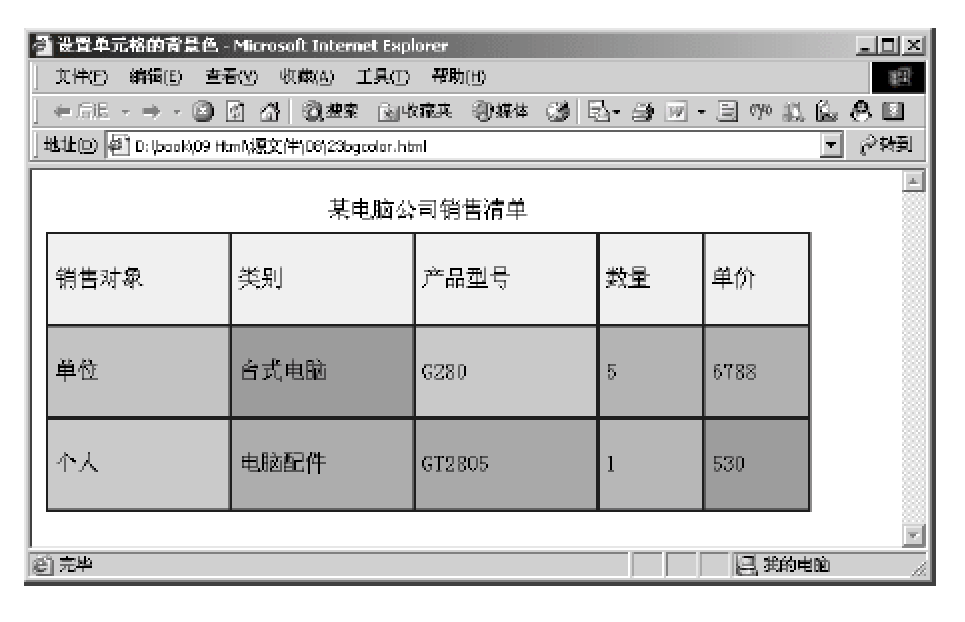


图 8-25　设置单元格的背景色

8.6.6　单元格的边框颜色——bordercolor

单元格的边框颜色可以通过 bordercolor 参数单独设置。

语法：<td bordercolor="颜色代码">

实例代码：

```
<head>
<title>设置单元格的边框颜色</title>
</head>
<body>
    <table border=2 bordercolor="#CCCCFF" cellspacing=5 cellpadding=5 height=140 width=550>
        <caption>某电脑公司销售清单</caption>
        <tr bgcolor="#EEFECC">
            <td>销售对象</td>
            <td>类别</td>
```

```
                <td>产品型号</td>
                <td>数量</td>
                <td>单价</td>
            </tr>
            <tr>
                <td bordercolor="#77EE00">单位</td>
                <td bordercolor="#BB8800">台式电脑</td>
                <td bordercolor="#7700CC">G280</td>
                <td bordercolor="#77CC88">5</td>
                <td bordercolor="#00AA77">6788</td>
            </tr>
            <tr>
                <td bordercolor="#8800FF">个人</td>
                <td bordercolor="#009970">电脑配件</td>
                <td bordercolor="#0088CC">GT2805</td>
                <td bordercolor="#B6B0E3">1</td>
                <td bordercolor="#BB8800">530</td>
            </tr>
        </table>
</body>
</html>
```

运行代码，会发现各个单元格的边框设置了不同的颜色，如图 8-26 所示。

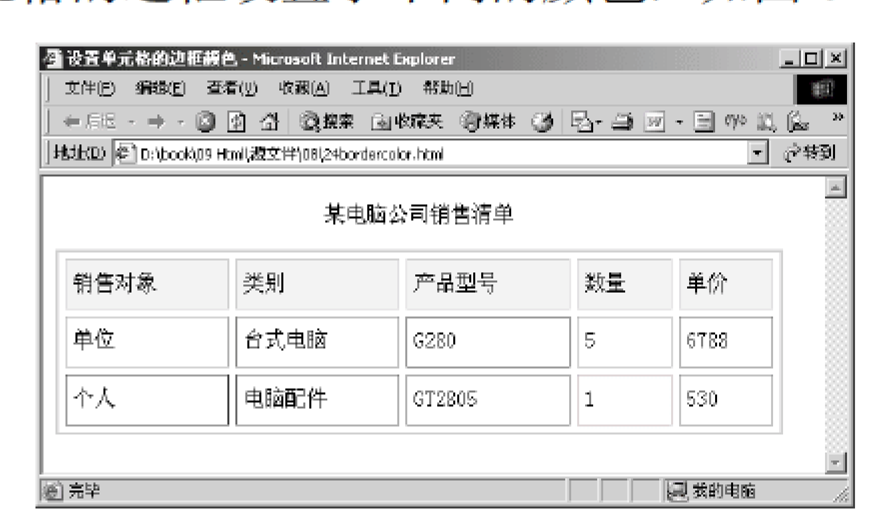


图 8-26　设置单元格边框颜色

8.6.7　设置单元格的亮边框——bordercolorlight

单元格的亮边框就是单元格边框向光的部分。通过 bordercolorlight 参数可以单独定义单元格亮边框的颜色。

语法：<tb bordercolorlight="颜色代码">

实例代码：

```
<head>
<title>设置单元格的亮边框颜色</title>
</head>
<body>
    <table border=2 bordercolor="#DDAAFF" cellspacing=5 height=140 width=550>
        <caption>某商店物价表</caption>
        <tr bgcolor="#EEFECC">
```

```
            <td>产品名称</td>
            <td>类别</td>
            <td>生产日期</td>
            <td>价格</td>
            <td>产地</td>
        </tr>
        <tr>
            <td bordercolorlight="#FF0000">苹果罐头</td>
            <td>食品</td>
            <td bordercolorlight="#0000FF">2005-7-22</td>
            <td>8.70</td>
            <td>上海</td>
        </tr>
        <tr>
            <td>红鹰钢笔</td>
            <td>日用品</td>
            <td bordercolorlight="#00FFEE">2005-2-15</td>
            <td>35.00</td>
            <td>北京</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，看到表格中设置了亮边框的效果，如图 8-27 所示。

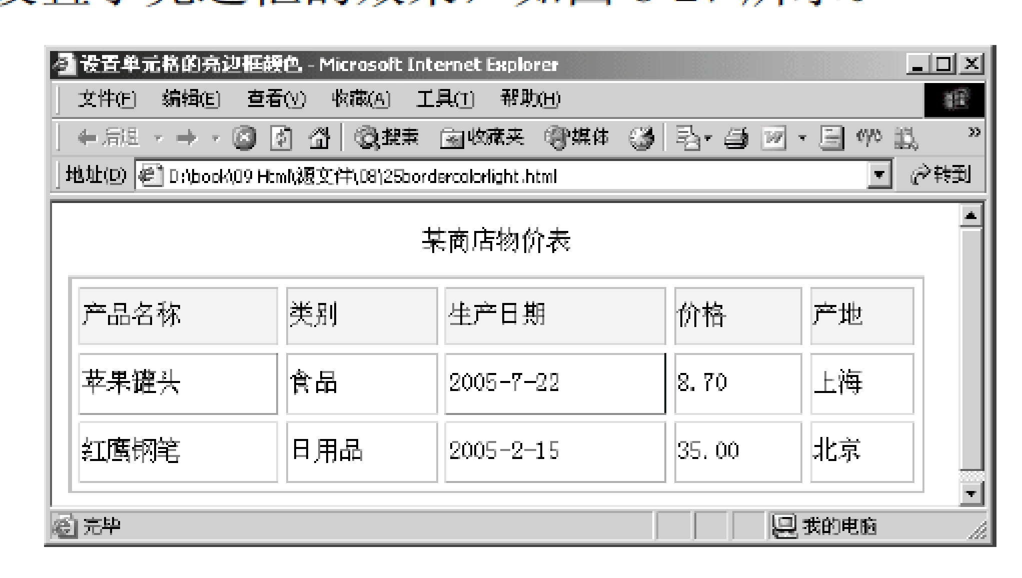


图 8-27　设置单元格的亮边框

8.6.8　设置单元格的暗边框——bordercolordark

单元格的暗边框就是单元格边框背光的部分。通过 bordercolordark 参数可以单独定义单元格暗边框的颜色。

语法：<tb bordercolordark="颜色代码">

实例代码：

```
<head>
<title>设置单元格的暗边框颜色</title>
</head>
<body>
    <table border=2 bordercolor="#DDAAFF" cellspacing=5 height=140 width=550>
```

```
        <caption>某商店物价表</caption>
        <tr bgcolor="#EEFECC">
            <td>产品名称</td>
            <td>类别</td>
            <td>生产日期</td>
            <td>价格</td>
            <td>产地</td>
        </tr>
        <tr>
            <td bordercolorlight ="#FF0000" bordercolordark ="#000099">苹果罐头</td>
            <td>食品</td>
            <td bordercolorlight="#0000FF" bordercolordark ="#000000">2005-7-22</td>
            <td>8.70</td>
            <td>上海</td>
        </tr>
        <tr>
            <td>红鹰钢笔</td>
            <td>日用品</td>
            <td bordercolorlight="#00FFEE" bordercolordark ="#330000">2005-2-15</td>
            <td>35.00</td>
            <td>北京</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，可以看到表格中设置了暗边框的效果，如图 8-28 所示。

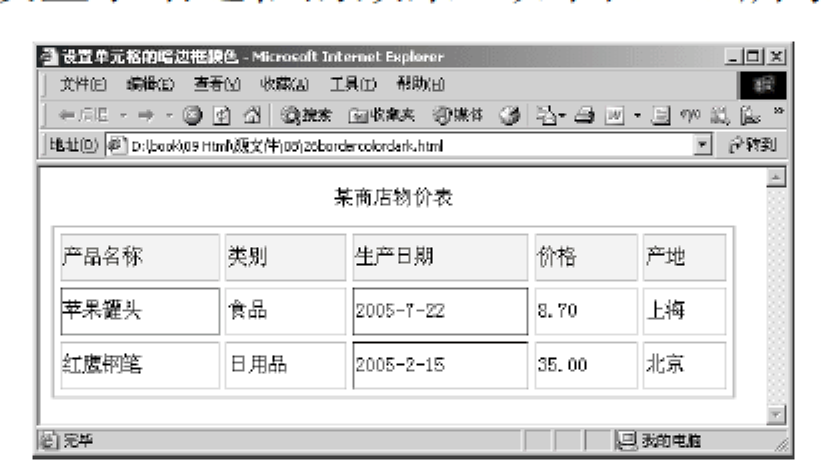


图 8-28　设置单元格的暗边框

8.6.9　设置单元格的背景图像——background

与表格的行设置不同的是，单元格可以设置背景为图像格式。

语法：<td background="背景图片的地址">

说明：背景图片的地址可以是绝对地址，也可以是相对地址。

实例代码：

```
<head>
<title>设置单元格的背景图片</title>
</head>
<body>
```

```
    <table border=2 bordercolor="#DDAAFF" cellspacing=5 height=140 width=550>
        <caption>某公司职员表</caption>
        <tr bgcolor="#FEDF9E">
            <td>职工代码</td>
            <td>部门</td>
            <td>姓名</td>
            <td>性别</td>
            <td>年龄</td>
            <td>职工类别</td>
        </tr>
        <tr>
            <td background="pic02.jpg">ZHGS01001</td>
            <td>总公司</td>
            <td bgcolor="#67AEFE">崔维军</td>
            <td>男</td>
            <td>35</td>
            <td>行政</td>
        </tr>
        <tr>
            <td>ZHGS01002</td>
            <td>总公司</td>
            <td>官莉</td>
            <td>女</td>
            <td>33</td>
            <td bgcolor=" #F5C1A5">行政</td>
        </tr>
        <tr>
            <td bgcolor=" #BC9CFE" background="pic02.jpg">ZHGS02001</td>
            <td>总公司</td>
            <td>费文龙</td>
            <td>男</td>
            <td>41</td>
            <td>技术</td>
        </tr>
        <tr>
            <td>ZHGS02002</td>
            <td>总公司</td>
            <td>顾炳华</td>
            <td>男</td>
            <td>27</td>
            <td>技术</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，效果如图 8-29 所示。当同时设置了单元格背景色和单元格背景图片时，网页中将会显示图片的效果，如第 4 行第 1 个单元格。



图 8-29　设置单元格的背景图像

8.6.10　文字内容不换行——nowrap

当表格没有设定宽度时，整个表的宽度会根据表格内容进行调整，但表格的宽度不会超出浏览器的宽度。当单元格内的内容过长时会自动换行，下面以实例说明。

实例代码：

```
<html>
<head>
<title>表格内容过长</title>
</head>
<body>
    <table align="center" border=1 cellspacing=0 bordercolor="#990000" cellpadding=3>
        <caption>通讯录</caption>
        <tr>
            <th>姓名</th>
            <th>地址</th>
            <th>电话</th>
            <th>电子邮件</th>
            <th>其他联系方法</th>
        </tr>
        <tr>
            <td>崔维</td>
            <td>天津市河东区王子胡同 183 号</td>
            <td>022-83546675</td>
            <td>ddzz@yahoo.com.cn</td>
            <td>13057891234</td>
        </tr>
        <tr>
            <td>冯炳华</td>
            <td>北京市景山公园 3 号楼 5 单元 1106 室</td>
            <td>010-63546874</td>
            <td>fengg165@jingshan.net</td>
            <td>13656462636</td>
        </tr>
        <tr>
            <td>王奇安</td>
            <td>南京市雨花路 875 号</td>
            <td>0352-87457454</td>
            <td>wqa@163.com</td>
            <td>13211118888</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，可以看到单元格的内容自动换行，效果如图 8-30 所示。



图 8-30　表头过长时换行显示

如果不希望表格中某个单元格的内容换行，可以通过 nowrap 参数进行设置。

语法：

<th nowrap>

<td nowrap>

说明：这一参数可以设置在表格标题中，也可以设置在普通单元格中。

实例代码：

```
<html>
<head>
<title>表格内容不换行</title>
</head>
<body>
    <table align="center" border=1 cellspacing=0 bordercolor="#990000" cellpadding=3>
        <caption>通讯录</caption>
        <tr>
            <th>姓名</th>
            <th>地址</th>
            <th>电话</th>
            <th>电子邮件</th>
            <th nowrap>其他联系方法</th>
        </tr>
        <tr>
            <td>崔维</td>
            <td nowrap>天津市河东区王子胡同 183 号</td>
            <td>022-83546675</td>
            <td>ddzz@yahoo.com.cn</td>
            <td>13057891234</td>
        </tr>
        <tr>
            <td nowrap >冯炳华</td>
            <td>北京市景山公园 3 号楼 5 单元 1106 室</td>
            <td>010-63546874</td>
            <td>fengg165@jingshan.net</td>
            <td>13656462636</td>
        </tr>
        <tr>
            <td nowrap >王奇安</td>
            <td>南京市雨花路 875 号</td>
            <td>0352-87457454</td>
            <td>wqa@163.com</td>
            <td>13211118888</td>
        </tr>
    </table>
</body>
</html>
```

运行代码，可以看到表格中设置了 nowrap 参数的单元格内容不换行显示了，如图 8-31 所示。而超出浏览器宽度的内容则通过浏览器的滚动条来显示。

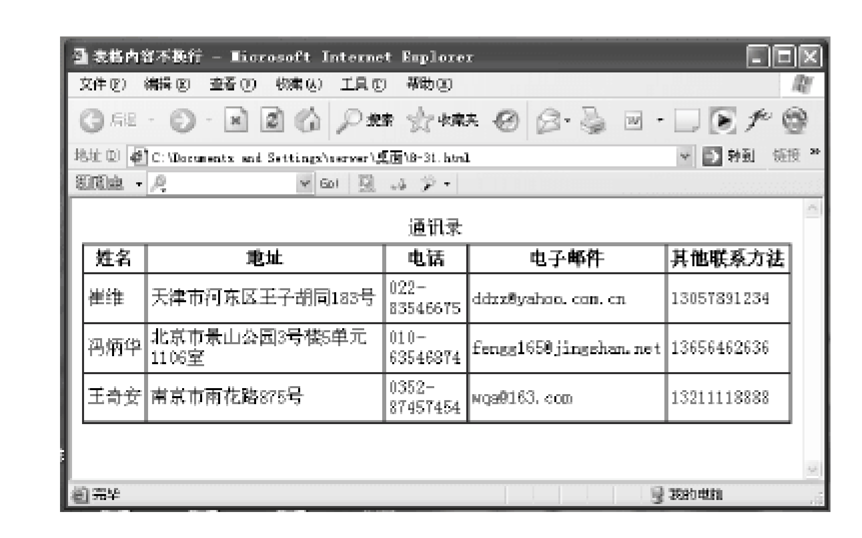


图 8-31　设置表格内容不换行

8.7　表格的结构

除了对表格的设计标记外，还有一些标记是用来明确表格结构的。通过对结构的设置而分别对表首、表主体以及表尾的样式进行设置。这些都通过成对出现的标记设置，应用到表格里用于整体规划表格的行列属性。使用这些标记能对表格的一行或多行单元格属性进行统一修改，从而省去了逐一修改单元格属性的麻烦。

8.7.1　设计表头样式——thead

表头样式的开始标记是<thead>，结束标记是</thead>。它们用于定义表格最上端表首的样式，其中可以设置背景颜色、文字对齐方式、文字的垂直对齐方式等。

语法：

```
<thead bgcolor="颜色代码" align="对齐方式" valign="垂直对齐方式">
    ……
</thead>
```

说明：在该语法中，bgcolor、align、valign 参数的取值范围与单元格中的设置方法相同，align 可以取值 left、center 或 right；valign 可以取值 top、middle 或 bottom。在<thead>标记内还可以包含<td>、<th>和<tr>标记，而一个表元素中只能有一个<thead>标记。

实例代码：

```
<html>
<head>
<title>设置表头的样式</title>
</head>
<body>
    <table align="center" border=1 bordercolor="#990000" cellpadding=3 width=550 height=180>
        <caption>某单位物品领用表</caption>
      <thead bgcolor="#DDDDFF" align="center" valign="bottom">
        <tr>
            <th>物品名</th>
            <th>类型</th>
            <th>领用人</th>
            <th>领用人部门</th>
            <th>数量</th>
        </tr>
      </thead>
        <tr>
            <td>网站管理与设计</td>
            <td>书籍</td>
            <td>王文</td>
            <td>技术开发部</td>
            <td>1</td>
```

```
        </tr>
        <tr>
            <td>XX 牌鼠标</td>
            <td>计算机设备</td>
            <td>李颖</td>
            <td>行政办公室</td>
            <td>1</td>
        </tr>
        <tr>
            <td>打印纸</td>
            <td>办公耗材</td>
            <td>田藤原</td>
            <td>售后服务部</td>
            <td>5</td>
        </tr>
    </table>
</body>
</html>
```

运行这段代码，效果如图 8-32 所示。

图 8-32　设计表头样式

8.7.2　设计表主体样式——tbody

与表头样式的标记功能类似，表主体样式用来统一设计表主体部分的样式，标记为<tbody>。

语法：

```
<tbody bgcolor="颜色代码" align="对齐方式" valign="垂直对齐方式">
    ……
</tbody>
```

说明：在该语法中，bgcolor、align、valign 参数的取值范围与<thead>标记中的相同。一个表元素中只能有一个<tbody>标记。

实例代码：

```
<html>
<head>
<title>设置表主体的样式</title>
</head>
<body>
    <table align="center" border=1 bordercolor="#990000" cellpadding=3 width=550 height=180>
        <caption>某单位物品领用表</caption>
      <thead bgcolor="#97B6FF" align="center" valign="bottom">
        <tr>
            <th>物品名</th>
            <th>类型</th>
            <th>领用人</th>
            <th>领用人部门</th>
            <th>数量</th>
        </tr>
      </thead>
```

```
        <tbody bgcolor="#FFF0D7" align="left" valign="bottom">
          <tr>
              <td>网站管理与设计</td>
              <td>书籍</td>
              <td>王文</td>
              <td>技术开发部</td>
              <td>1</td>
          </tr>
          <tr>
              <td>XX 牌鼠标</td>
              <td>计算机设备</td>
              <td>李颖</td>
              <td>行政办公室</td>
              <td>1</td>
          </tr>
          <tr>
              <td>打印纸</td>
              <td>办公耗材</td>
              <td>田藤原</td>
              <td>售后服务部</td>
              <td>5</td>
          </tr>
        </tbody>
      </table>
</body>
</html>
```

运行这段代码，可以看到表格的主体内容统一设置了样式，如图 8-33 所示。

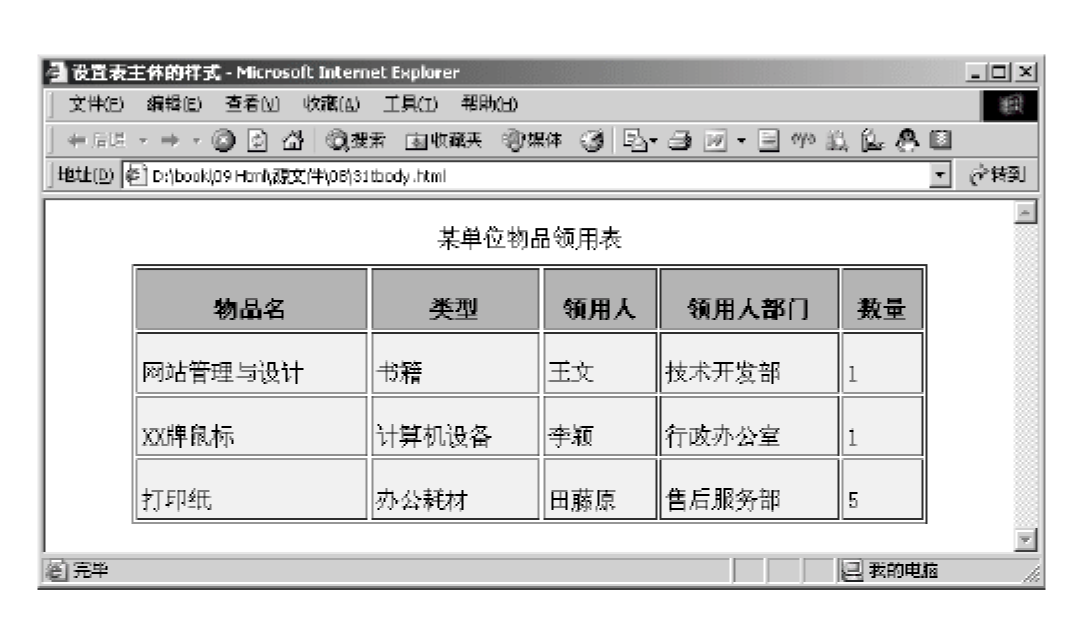


图 8-33　设置表格主体的样式

8.7.3　设计表尾样式——tfoot

<tfoot>标记用于定义表尾样式。

语法：

```
<tfoot bgcolor="颜色代码" align="对齐方式" valign="垂直对齐方式">
    ……
</tfoot>
```

说明：在该语法中，bgcolor、align、valign 参数的取值范围与<thead>标记中的相同。一个表元素中只能有一个<tfoot>标记。

实例代码：

```
<html>
<head>
<title>设置表尾的样式</title>
</head>
<body>
```

```
    <table align="center" border=1 bordercolor="#990000" cellpadding=3 width=550 height=160>
        <caption>某单位物品领用表</caption>
      <thead bgcolor="#97B6FF" align="center" valign="bottom">
        <tr>
              <th>物品名</th>
              <th>类型</th>
              <th>领用人</th>
              <th>领用人部门</th>
              <th>数量</th>
        </tr>
      </thead>
      <tbody bgcolor="#FFF0D7" align="left" valign="bottom">
        <tr>
              <td>网站管理与设计</td>
              <td>书籍</td>
              <td>王文</td>
              <td>技术开发部</td>
              <td>1</td>
        </tr>
        <tr>
              <td>XX 牌鼠标</td>
              <td>计算机设备</td>
              <td>李颖</td>
              <td>行政办公室</td>
              <td>1</td>
        </tr>
        <tr>
              <td>打印纸</td>
              <td>办公耗材</td>
              <td>田藤原</td>
              <td>售后服务部</td>
              <td>5</td>
        </tr>
      </tbody>
      <tfoot bgcolor="#FAA9AB" align="right" valign="middle">
        <tr>
              <td colspan=5>表格创建日期：XX-XX-XX</td>
        </tr>
      </tfoot>
    </table>
</body>
</html>
```

运行这段代码，效果如图 8-34 所示。

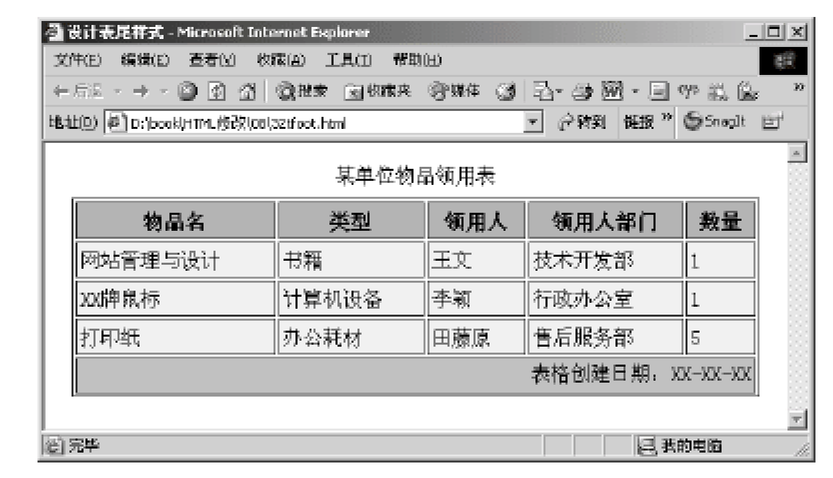


图 8-34　设计表尾样式

8.8　表格的嵌套

在实际应用中，表格并不是单一出现的，往往需要在表格内嵌套其他的表格来实现页面的整体布局。一般情况下需要使用一些可视化软件来实现布局，这样看起来比较直观，容易达到预期的目的。也可以直接通过输入代码来实现。下面举例说明表格的嵌套。

实例代码：

```
<html>
<head>
<title>表格的嵌套</title>
</head>
<body>
<!--使用表格的嵌套功能设计网页的版式-->
<table width="560" height="300" border="1" cellspacing="0" align="center">
<thead bgcolor="#8A84FF">
    <tr height="70">
        <td width="160">网站 logo</td>
        <td width="400">网站 banner</td>
    </tr>
</thead>
<tbody>
    <tr valign="top" height="200">
        <td width="160" align="center">
        <table width="135" height="180" border="1" cellspacing="0" bgcolor="#97B6FF">
            <tr>
                <td>页面导航</td>
            </tr>
            <tr>
                <td>页面导航</td>
            </tr>
            <tr>
                <td>页面导航</td>
            </tr>
            <tr>
                <td>页面导航</td>
            </tr>
            <tr>
                <td>页面导航</td>
            </tr>
        </table>
        </td>
        <td width="400" height="200" background="pic03.jpg">
        <table width="380" height="160" border="1" bordercolor="#FF9900" cellspacing="2" cellpadding="5">
            <tr>
                <td>网站板块</td>
                <td>网站板块</td>
```

```
                </tr>
                <tr>
                    <td>网站板块</td>
                    <td>网站板块</td>
                </tr>
            </table>
            </td>
        </tr>
    </tbody>
    <tfoot bgcolor="#0000FF">
        <tr align="center">
            <td height="30" colspan="2"><font color="#FFFFFF">版权信息</font></td>
        </tr>
    </tfoot>
    </table>
    </body>
    </html>
```

运行这段代码，可以看到设计的网页版式如图 8-35 所示。

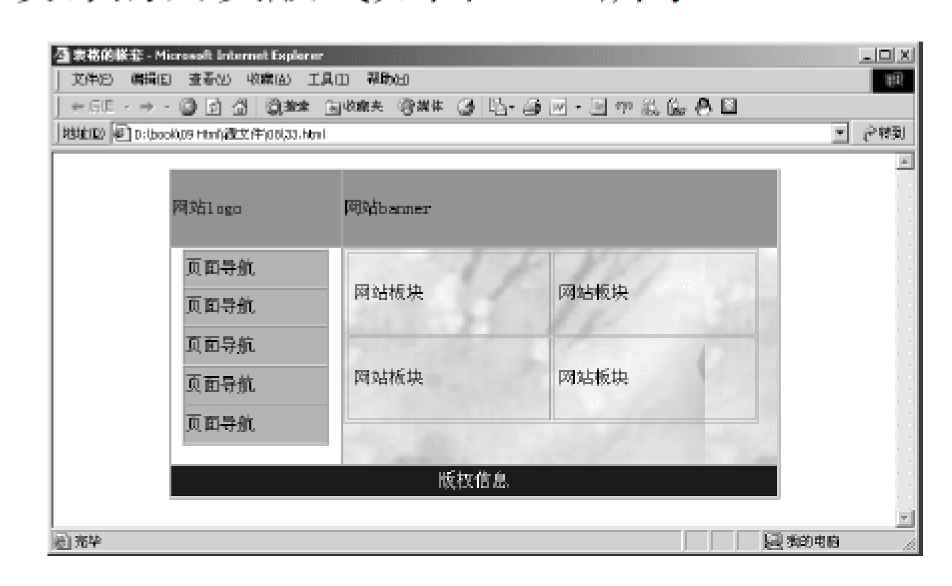


图 8-35　表格的嵌套效果

8.9　层标记——div

层标记也可以称为区隔标记，可以看作是为网页排版的标记。在这一方面它与表格有着相似的功能，但层能够完成更加复杂、更加灵活的排版效果。它能够将字、画、表格等多种元素组成一个区域进行样式的统一设置。

语法：<div id=值 align=对齐方式 style=样式 class=应用的样式类></div>

说明：在该语法中，id 用来标识层；align 用来设定层内元素的对齐方式，包括左对齐、右对齐和居中对齐；style 则用来设定层的属性，包括层的大小范围和起始位置；class 用于定义层所应用的 CSS 样式。一般情况下，div 标记常常和 CSS 样式表结合使用，这在后面讲解 CSS 样式时，会充分用到 div 标记将一部分页面元素组合成一个块级区域。

实例代码：

```
<html>
<head>
<title>设置层</title>
```

```
</head>
<body>
      <div id="exam1" align="center">
        <h3>轩辕剑三外传：天之痕</h3>
        <hr size=2>
        <p>神州大地上，从神话时代流传下来十种上古神器——钟、剑、斧、壶、塔、琴、鼎、印、镜、石。它们各自有着迥然不同的绝世力量。只要稍加利用即可纵横四海，无敌天下。但它们的下落，已湮灭于神州漫长之乱世历史中。
        <p>除了轩辕剑，还有创世神开天辟地使用的神器炼妖壶，在上古英雄的手中辗转流传，在这些古人的庇佑下，中国到了文化鼎盛的时代——隋唐。
        <p>公元七世纪，南北朝时期，北朝皇帝隋文帝派兵消灭了南朝陈国后，结束了中原南北朝数百年之分裂局面，重新统一了中国并成立了隋国。然而陈国的遗民悲痛祖国亡灭，在江南集结大军数万兴兵反抗，企图一举复国。隋文帝下令出兵平乱，然而最令人惊讶的是，这支平乱部队为首竟是一位身披神秘斗篷，年近十二岁的少年。他手持一把雕有古朴花纹的金色之剑，此乃十大上古神器之一——轩辕剑。少年在一击之间，便将反抗朝廷的陈国数万人马化为飞灰！自此，大隋威名让所有反抗者闻之色变，再没有人敢轻起反抗之念。
        <p>天之痕的故事从十六年后开始，发生在轩辕剑三的一百三十三年前……
      </div>
      <div id="exam2" style="position:absolute;top:70pt;left:170pt;">
        <img src="pic04.jpg" width=220>
      </div>
<br>
</body>
</html>
```

运行程序，效果如图 8-36 所示。其中第 1 个层将文字元素放置其中，并设置对齐方式为居中对齐；而第 2 个层中放置了一个图片，并使用 style 参数设定了该层的位置。style 参数中取值的具体设置方法在后面 CSS 样式表一章（第 11 章）中还将具体介绍，这里只需知道设置了层的位置即可。

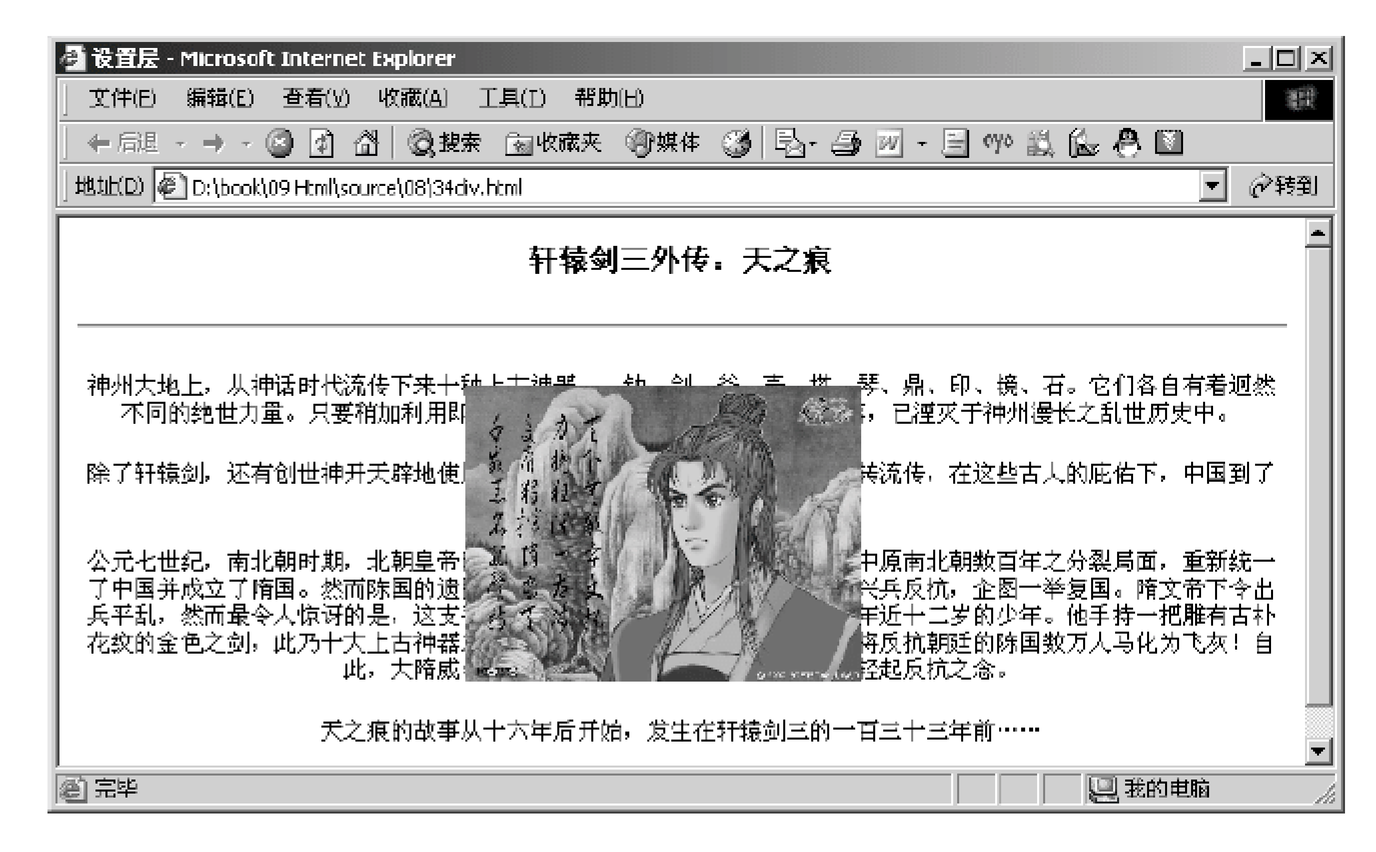


图 8-36　层的效果

第 9 章 添加表单

- 表单标记——form
- 添加控件
- 输入类的控件
- 菜单列表类的控件
- 文本域标记——textarea
- id 标记

表单的用途很多，在制作网页，特别是制作动态网页时常常会用到。表单主要用来收集客户端提供的相关信息，使网页具有交互的功能。它是 HTML 页面与浏览器实现交互的重要手段。在网页的制作过程中，常常需要使用表单，例如在进行用户注册时，就必须通过表单填写用户的相关信息。本章将讲解各种表单的用法。

表单通常设计在一个 HTML 文档中，当用户填写完信息后做提交操作，将表单的内容从客户端的浏览器传送到服务器上，经过服务器处理程序后，再将用户所需信息传送回客户端的浏览器上，这样网页就具有了交互性。

在网页中，最常见的表单形式主要包括文本框、单选按钮、复选框、按钮等，如图 9-1 所示。在网易的主页中，就包含了文本框、按钮、下拉列表等表单内容。

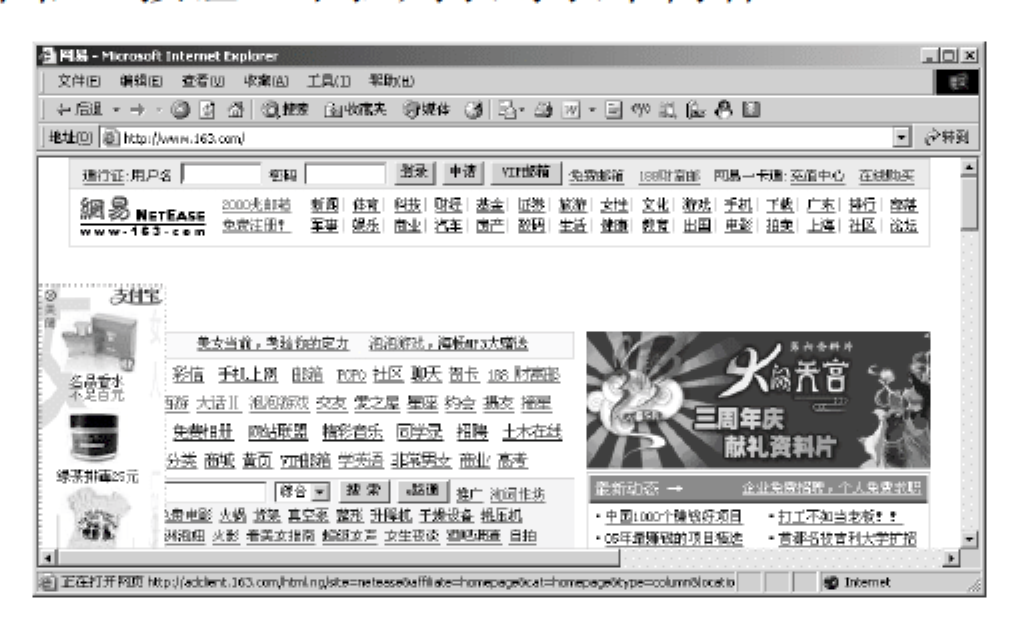


图 9-1　网页中的表单

下面就对如何使用 HTML 标志来设计表单进行详细的介绍。

9.1　表单标记——form

在 HTML 中，<form></form>标志对用来创建一个表单，也即定义表单的开始和结束位置，在标志对之间的一切都属于表单的内容。

每个表单元素开始于 form 元素，可以包含所有的表单控件，还有任何必需的伴随数据，如控件的标签、处理数据的脚本或程序的位置等。在表单的<form>标记中，还可以设置表单的基本属性，包括表单的名称、处理程序、传送方法等。一般情况下，表单的处理程序 action 和传送方法 method 是必不可少的参数。

9.1.1　处理程序——action

真正处理表单的数据脚本或程序在 action 属性里，这个值可以是程序或脚本的一个完整 URL。

语法：

```
<form action="表单的处理程序">
    ……
</form>
```

说明：在该语法中，表单的处理程序定义的是表单要提交的地址，也就是表单中收集到的资料将要传递的程序地址。这一地址可以是绝对地址，也可以是相对地址，还可以是一些其他的地址形式，例如发送 E-mail 等。

实例代码：

```
<html>
```

```
<head>
<title>设定表单的处理程序</title>
</head>
<body>
    下面是关于本产品的调查内容：
    <!--这是一个没有控件的表单-->
    <form action="mailto:abcd@163.com">
    </form>
</body>
</html>
```

在这个实例中，就是定义了表单提交的地址为一个邮件，当程序运行后会将表单中收集到的内容以电子邮件的形式发送出去。

9.1.2　表单名称——name

名称属性 name 用于给表单命名。这一属性不是表单的必需属性，但是为了防止表单信息在提交到后台处理程序时出现混乱，一般要设置一个与表单功能符合的名称，例如注册页面的表单可以命名为 register。不同的表单尽量不用相同的名称，以避免混乱。

语法：

```
<form name="表单名称">
    ……
</form>
```

说明：表单名称中不能包含特殊符号和空格。

实例代码：

```
<html>
<head>
<title>设定表单的名称</title>
</head>
<body>
    下面是关于本产品的调查内容：
    <!--这是一个没有控件的表单-->
    <form action="mailto:abcd@163.com" name="research">
    </form>
</body>
</html>
```

在该实例中，将表单命名为 research。

9.1.3　传送方法——method

表单的 method 属性用来定义处理程序从表单中获得信息的方式，可取值为 get 或 post，它决定了表单中已收集的数据是用什么方法发送到服务器的。

- method=get：使用这个设置时，表单数据会被视为CGI或ASP的参数发送，也就是来访者输入

的数据会附加在URL之后，由用户端直接发送至服务器，所以速度上会比post快，但缺点是数据长度不能够太长。在没有指定method的情形下，一般都会视get为默认值。

- method=post：使用这种设置时，表单数据是与URL分开发送的，用户端的计算机会通知服务器来读取数据，所以通常没有数据长度上的限制，缺点是速度上会比get慢。

语法：

```
<form method="传送方式">
    ……
</form>
```

说明：传送方式的值只有两种选择即 get 或 post。

实例代码：

```
<html>
<head>
<title>设定表单的传送方式</title>
</head>
<body>
    下面是关于本产品的调查内容：
    <!--这是一个没有控件的表单-->
    <form action="mailto:abcd@163.com" name="research" method="post">
    </form>
</body>
</html>
```

在这个实例里，表单 research 的内容将会以 post 的方式通过电子邮件的形式传送出去。

9.1.4 编码方式——enctype

表单中的 enctype 参数用于设置表单信息提交的编码方式。

语法：

```
<form enctype="编码方式">
    ……
</form>
```

说明：enctype 属性为表单定义了 MIME 编码方式，编码方式的取值见表 9-1。

表9-1　目标窗口的设置

enctype取值	取值的含义
Text/plain	以纯文本的形式传送
application /x-www-form-urlencoded	默认的编码形式
multipart/form-data	MIME编码，上传文件的表单必须选择该项

实例代码：

```
<html>
<head>
```

```
<title>设定表单的编码方式</title>
</head>
<body>
    下面是关于本产品的调查内容：
    <!--这是一个没有控件的表单-->
    <form action="mailto:abcd@163.com" name="research" method="post" enctype="Text/plain">
    </form>
</body>
</html>
```

在这个实例中，设置了表单信息以纯文本的编码形式发送。

9.1.5　目标显示方式——target

target 属性用来指定目标窗口的打开方式。表单的目标窗口往往用来显示表单的返回信息，例如是否成功提交了表单的内容、是否出错等。

语法：

```
<form target="目标窗口的打开方式">
    ……
</form>
```

说明：目标窗口的打开方式包含 4 个取值：_blank、_parent、_self 和_top。其中_blank 是指将返回的信息显示在新打开的窗口中；_parent 是指将返回信息显示在父级的浏览器窗口中；_self 则表示将返回信息显示在当前浏览器窗口；_top 表示将返回信息显示在顶级浏览器窗口中。

实例代码：

```
<html>
<head>
<title>设定表单的目标窗口打开方式</title>
</head>
<body>
    下面是关于本产品的调查内容：
    <!--这是一个没有控件的表单-->
   <formaction="mailto:abcd@163.com" name="research" method="post" enctype="Text/plain"target="_self">
    </form>
</body>
</html>
```

在这个实例中，设置表单的返回信息将在同一窗口中显示。

以上所讲解的只是表单的基本构成标记，而表单的<form>标记只有和它所包含的具体控件相结合才能真正实现表单收集信息的功能。下面就对表单中各种功能的控件的添加方法加以说明。

9.2　添 加 控 件

按照控件的填写方式可以分为输入类和菜单列表类。输入类的控件一般以 input 标记开始，说明这一控件需要用户的输入；而菜单列表类则以 select 开始，表示用户需要选择。按照控件的表现形式则可以将控件分为文本类、选项按钮、菜单等几种。

在 HTML 表单中，input 参数是最常用的控件标记，包括最常见的文本域、按钮都是采用这个标记。这个标记的基本语法是：

```
<form>
    <input name="控件名称" type="控件类型">
</form>
```

在这里，控件名称是为了便于程序对不同控件的区分，而 type 参数则是确定了这一个控件域的类型。在 HTML 中，input 参数所包含的控件类型可以见表 9-2。

表9-2　输入类控件的Type可选值

type取值	取值的含义
text	文字字段
password	密码域，用户在页面中输入时不显示具体的内容，以*代替
radio	单选按钮
checkbox	复选框
button	普通按钮
submit	提交按钮
reset	重置按钮
image	图形域，也称为图像提交按钮
hidden	隐藏域，隐藏域将不显示在页面上，只将内容传递到服务器中
file	文件域

除了输入类型的控件之外，还有一些控件，如文字区域、菜单列表则不是用 input 标记的。它们有自己的特定标记，如文字区域直接使用 textarea 标记，菜单标记需要使用 select 和 option 标记结合，这些在后面还将详细介绍。

9.3　输入类的控件

9.3.1　文字字段——text

在网页中最常见的就是文本字段的表单，例如网页的用户登录区。文字字段的 type 属性值为 text，而 text 类型的控件在页面中以单行文本框的形式显示，在页面中还可以设置控件的名称、控件的长度、

最长字符数等。

语法：<input type="text" name="控件名称" size=控件的长度 maxlength=最长字符数 value="文字字段的默认取值">

说明：在该语法中包含了很多参数，它们的含义和取值方法不同，见表 9-3。其中 name、size、maxlength 参数一般是不会省略的参数。

表9-3 text文字字段的参数表

参 数 类 型	含 义
name	文字字段的名称，用于和页面中其他控件加以区别，命名时不能包含特殊字符，也不能以HTML预留字作为名称
size	定义文本框在页面中显示的长度，以字符作为单位
maxlength	定义在文本框中最多可以输入的文字数
value	用于定义文本框中的默认值

实例代码：

```
<html>
<head>
<title>在表单中添加文字字段</title>
</head>
<body>
    下面是几种不同属性的文字字段：
    <form name="example" action="deal.asp" method="post">
      <!--添加一个长度为 15 的文本框-->
      姓名：<input type="text" name="username" size=15>
      <br>
      <!--添加一个长度为 15，但是最长字符为 2 的文本框-->
      年龄：<input type="text" name="age" size=15 maxlength=2>
      <br>
      <!--添加一个长度为 15，但最多可输入 30 个字符，默认显示“http://”的文本框-->
      个人主页：<input type="text" name="privateweb" size=15 maxlength=30 value="http://">
    </form>
</body>
</html>
```

运行这段代码，可以看到几种不同大小的文字字段控件，如图 9-2 所示。

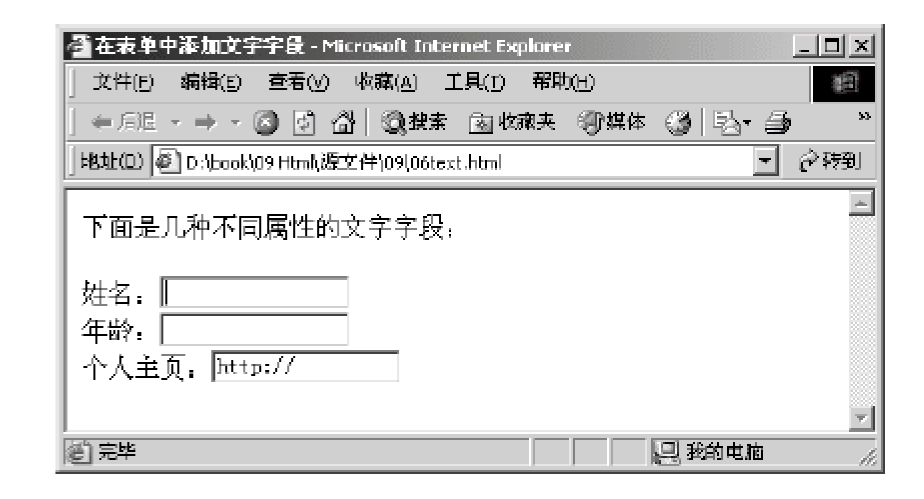


图 9-2 在页面中添加文字字段

9.3.2　密码域——password

在网页中有一种特殊的文本字段，它在页面中的效果和文本字段相同，但是当用户输入文字时，这些文字只显示“*”，这种控件称为密码域。

语法：<input type="password" name="控件名称" size=控件的长度 maxlength=最长字符数 value="文字字段的默认取值">

说明：在该语法中包含了很多参数，它们的含义和取值见表 9-4。其中 name、size、maxlength 参数一般是不会省略的参数。

表9-4　text文字字段的参数表

参 数 类 型	含　　义
name	域的名称，用于和页面中其他控件加以区别，命名时不能包含特殊字符，也不能以HTML预留字作为名称
size	定义密码域的文本框在页面中显示的长度，以字符作为单位
maxlength	定义在密码域的文本框中最多可以输入的文字数
value	用于定义密码域的默认值，同样以“*”显示

实例代码：

```
<html>
<head>
<title>在表单中添加密码域</title>
</head>
<body>
    下面是几种不同效果的密码域：
    <form name="example" action="deal.asp" method="post">
        <!--添加一个长度为 22 的密码域-->
        登录密码：<input type="password" name="username" size=22>
        <br>
        <!--添加一个长度为 22，但是最多可以输入 30 个字符的密码域-->
        支付密码：<input type="password" name="age" size=22 maxlength=30>
        <br>
        <!--添加一个长度为 22、最多可输入 30 个字符、默认密码设置为 12345 的密码域-->
        原始密码：<input type="password" name="privateweb" size=22 maxlength=30 value="12345">
    </form>
</body>
</html>
```

运行这段代码，可以看到几种不同大小的密码域控件，如图 9-3 所示。

在页面中输入文字 html，可以看到出现在文本框中的内容不是文字本身，而是 4 个星号“*”，如图 9-4 所示。

图 9-3　在页面中添加密码域



图 9-4　在密码域中输入文字

9.3.3　单选按钮——radio

在网页中，单选按钮用来让浏览者进行单一选择，在页面中以圆框表示。在单选按钮控件中必须设置参数 value 的值。而对于一个选择中的所有单选按钮来说，往往要设定同样的一个名称，这样在传递时才能更好地对某一个选择内容的取值进行判断。

语法：<input type="raido" value="单选按钮的取值" name="单选按钮名称" checked>

说明：在该语法中，checked 属性表示这一单选按钮默认被选中，而在一个单选按钮组中只能有一项单选按钮控件设置为 checked。value 则用来设置用户选中该项目后，传送到处理程序中的值。

实例代码：

```
<html>
<head>
<title>在表单中添加单选按钮</title>
</head>
<body>
      <h2>心理小测试：脱鞋方式见人心</h2>
      <hr>
      下班回家，在门口脱下了鞋子，从脱下鞋子的摆放形式，可以反映出你的性格的中心部分呢。试试看吧：
      <hr>
      <form name="example" action="deal.asp" method="post">
         <input type="radio" name="test" value="answerA" checked>鞋尖朝入口处排好
         <br>
         <input type="radio" name="test" value="answerB" >鞋尖朝进来的方向排好
         <br>
         <input type="radio" name="test" value="answerC" >就是脱掉的样子
         <br>
         <input type="radio" name="test" value="answerD" >由同住在
一起的人帮你脱
      </form>
</body>
</html>
```

运行程序，可以看到在页面中包含了 4 个单选按钮，如图 9-5 所示。

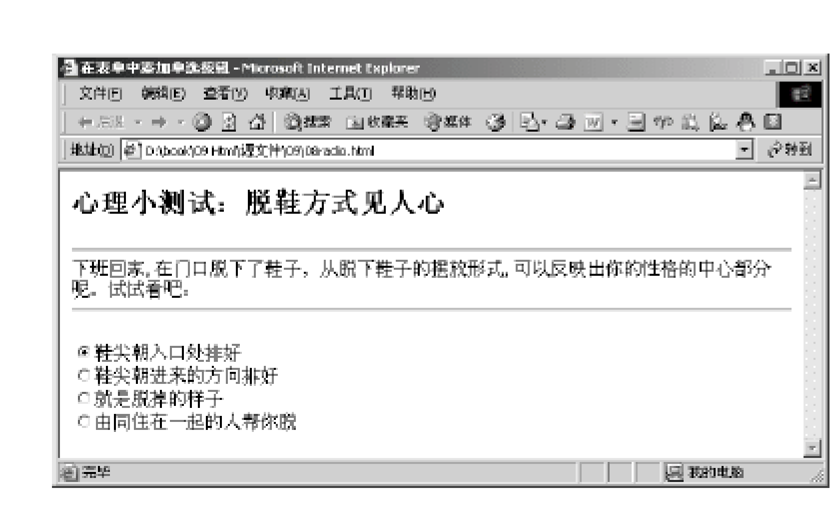


图 9-5　添加单选按钮

9.3.4　复选框——checkbox

在网页设计中，有一些内容需要让浏览者以选择的形式填写，而选择的内容可以是一个，也可以是多个，这时就需要使用复选框控件 checkbox。复选框在页面中以一个方框来表示。

语法：<input type="checkbox" value="复选框的值" name="名称" checked>

说明：在该语法中，checked 参数表示该选项在默认情况下已经被选中，一个选择中可以有多个复选框被选中。

实例代码：

```
<html>
<head>
<title>在表单中添加复选框</title>
</head>
<body>
    请在下面的选项中选择您喜欢的运动：
    <form name="example" action="deal.asp" method="post">
      <input type="checkbox" name="test" value="A2" checked>竞走
      <input type="checkbox" name="test" value="A3">体操
      <input type="checkbox" name="test" value="A1" checked>保龄
      <input type="checkbox" name="test" value="A4" >自行车
      <br>
      <input type="checkbox" name="test" value="A5" >登山
      <input type="checkbox" name="test" value="A6" >长跑
      <input type="checkbox" name="test" value="A7" >武术
      <input type="checkbox" name="test" value="A8" >游泳
      <br>
      <input type="checkbox" name="test" value="A11" >网球
      <input type="checkbox" name="test" value="A12" >篮球
      <input type="checkbox" name="test" value="A9" >排球
      <input type="checkbox" name="test" value="A10" >乒乓球
    </form>
</body>
</html>
```

运行代码，效果如图 9-6 所示。

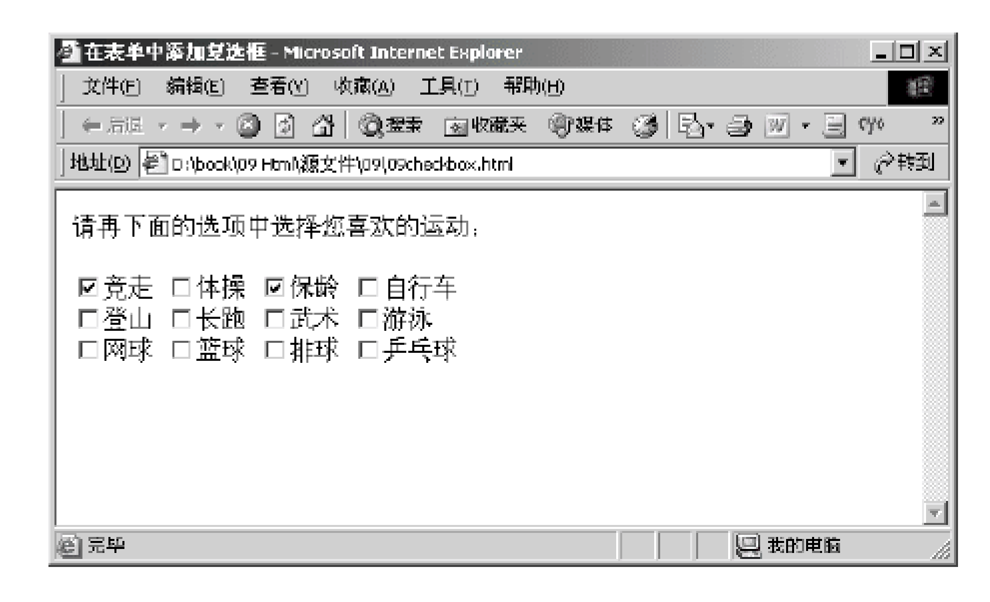


图 9-6　添加复选框的效果

9.3.5　普通按钮——button

在网页中按钮也很常见，在提交页面、恢复选项时常常用到。普通按钮一般情况下要配合脚本来进行表单处理。

语法：<input type="button" name="按钮名" value="按钮的取值" onclick="处理程序">

说明：value 的取值就是显示在按钮上面的文字，而在 button 中可以通过添加 onclick 参数来实现一些特殊的功能，onclick 参数是设置当鼠标按下按钮时所进行的处理。

实例代码：

```
<html>
<head>
<title>在表单中添加普通按钮</title>
</head>
<body>
    下面是几个有不同功能的按钮：<br><br>
    <form name="example" action="deal.asp" method="post">
      <!--在页面中添加一个普通按钮-->
      <input type="button" value="普通按钮" name="button1" >
      <!--在页面中添加一个关闭当前窗口的按钮-->
      <input type="button" name="close" value="关闭当前窗口" onclick="window.close()">
      <!--在页面中添加一个打开新窗口的按钮-->
      <input type="button" name="opennew" value="打开窗口" onclick="window.open()">
    </form>
</body>
</html>
```

运行这段代码，可以看到如图 9-7 所示的效果。

单击页面中的“普通按钮”按钮，页面不会有任何变化，因为在“普通按钮”按钮的代码中没有设置处理程序；如果单击“关闭当前窗口”按钮，会弹出一个关闭警告的窗口，如图 9-8 所示。

单击警告窗口中的“是（Y）”按钮，则会成功关闭当前窗口，否则返回。单击页面中的“打开窗口”按钮，会弹出一个新的窗口，如图 9-9 所示。



图 9-7　添加按钮



图 9-8　单击“关闭当前窗口”按钮后



图 9-9　打开新的窗口

9.3.6　提交按钮——submit

提交按钮是一种特殊的按钮，不需要设置 onclick 参数，在单击该类按钮时可以实现表单内容的提交。

语法：<input type="submit" name="按钮名" value="按钮的取值">

说明：在该语法中，value 同样用来设置按钮上显示的文字。

实例代码：

```
<html>
<head>
<title>添加提交按钮</title>
</head>
<body>
     下面是关于本网站的调查内容：
     <form action="mailto:abcd@163.com" name="research" method="post">
       <p>登录用户名：
         <input name="username" type="text" size=20>
       </p>
       <p>登录密码：
         <input name="password" type="password" size=20>
       </p>
        重复密码：
          <p>
         <input name="password2" type="password" size=20>
          </p>
     <hr>
       <p>调查的具体内容：</p>
       <p>所在城市：
          <input name="city" type="radio" value="radiobutton" checked>北京  
          <input name="city" type="radio" value="radiobutton">上海  
          <input name="city" type="radio" value="radiobutton">天津  
          <input name="city" type="radio" value="radiobutton">深圳  
          <input name="city" type="radio" value="radiobutton">其他 <input name="city" type="text">
       </p>
       <p>关心的内容：
          <input name="content" type="checkbox" value="checkbox" checked>新闻栏目   
          <input name="content" type="checkbox" value="checkbox">体育栏目   
          <input name="content2" type="checkbox" value="checkbox">科技内容   
          <input name="content3" type="checkbox" value="checkbox">动漫卡通<br>
                      
          <input name="content4" type="checkbox" value="checkbox">财经证券   
          <input name="content5" type="checkbox" value="checkbox">娱乐生活   
          <input name="content6" type="checkbox" value="checkbox">汽车房产   
          <input name="content7" type="checkbox" value="checkbox">出国旅游</p>
       <p>
         <input type="submit" name="Submit" value="提交">
       </p>
     </form>
</body>
</html>
```

由代码可知，本页面的信息将以电子邮件的形式传送。运行这段代码，效果如图 9-10 所示。

单击页面中的“提交”按钮，则会弹出一个打开如图 9-11 所示的提示窗口，提醒用户将会以电子

邮件的方式将表单内容传送出去。

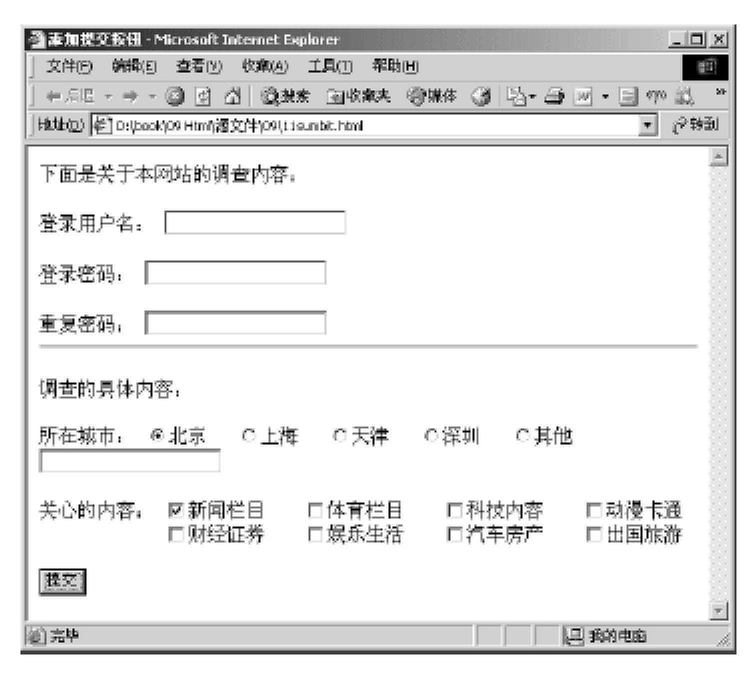


图 9-10　设置页面的效果



图 9-11　提交表单的提示窗口

9.3.7　重置按钮——reset

在页面中还有一种特殊的按钮，称为重置按钮。这类按钮可以用来清除用户在页面中输入的信息。

语法：<input type="reset" name="按钮名" value="按钮的取值">

说明：在该语法中，value 同样用来设置按钮上显示的文字。

实例代码：

```
<html>
<head>
<title>添加重置按钮</title>
</head>
<body>
     下面是某网站的注册页面：
     <form action="mailto:abcd@163.com" name="research" method="post">
       <p>用户名：
         <input name="username" type="text" size=20>
       </p>
       <p>登录密码：
         <input name="password" type="password" size=20>
     </p>
       <p>重复密码：
       <input name="password2" type="password" size=20>
        </p>
       <p>证件类型：
         <input name="papertype" type="text" size=20>
       </p>
       <p>证件号码：
         <input name="papernum" type="text" size=20 maxlength=35>
       </p>
       <p>出生日期：
         <input name="date" type="text" size=20>
       </p>
       <p>联系方式：
```

```
            <input name="touch" type="text" size=20 maxlength=50>
        </p>
        <p>所在城市：
            <input name="city" type="radio" value="radiobutton" checked>北京  
            <input name="city" type="radio" value="radiobutton">上海  
            <input name="city" type="radio" value="radiobutton">天津<br>
                      
            <input name="city" type="radio" value="radiobutton">深圳  
            <input name="city" type="radio" value="radiobutton">其他 <input name="city" type="text">
        </p>
        <p>关心的栏目：
            <input name="content" type="checkbox" value="checkbox" checked>新闻栏目   
            <input name="content" type="checkbox" value="checkbox">体育栏目   
            <input name="content2" type="checkbox" value="checkbox">科技内容<br>
                        
            <input name="content3" type="checkbox" value="checkbox">动漫卡通   
            <input name="content4" type="checkbox" value="checkbox">财经证券   
            <input name="content5" type="checkbox" value="checkbox">娱乐生活<br>
                        
            <input name="content6" type="checkbox" value="checkbox">汽车房产   
      <input name="content7" type="checkbox" value="checkbox">出国旅游
        </p>
        <p>
          <input type="submit" name="Submit" value="提交">
                     
          <input type="reset" name="reset" value="重置">
        </p>
      </form>
</body>
</html>
```

运行代码，效果如图 9-12 所示。

在浏览器中添加部分信息后，效果如图 9-13 所示。单击“重置”按钮，会发现填写页面中的内容全部被清除，又恢复到了图 9-12 所示的效果。

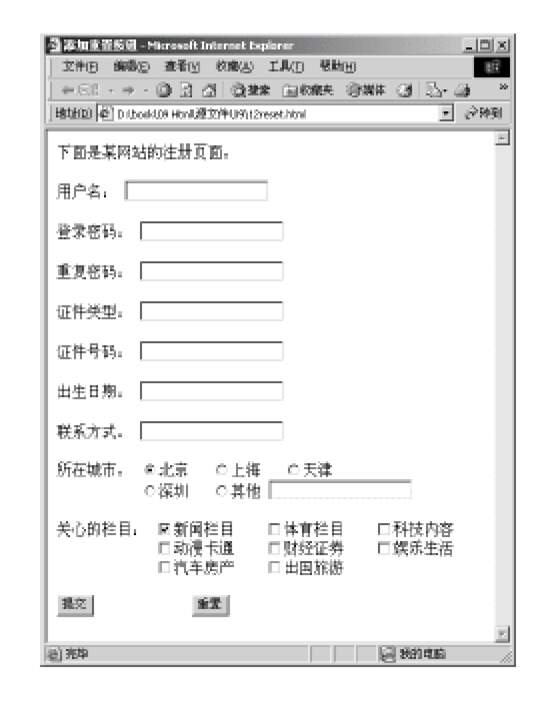


图 9-12　添加重置按钮

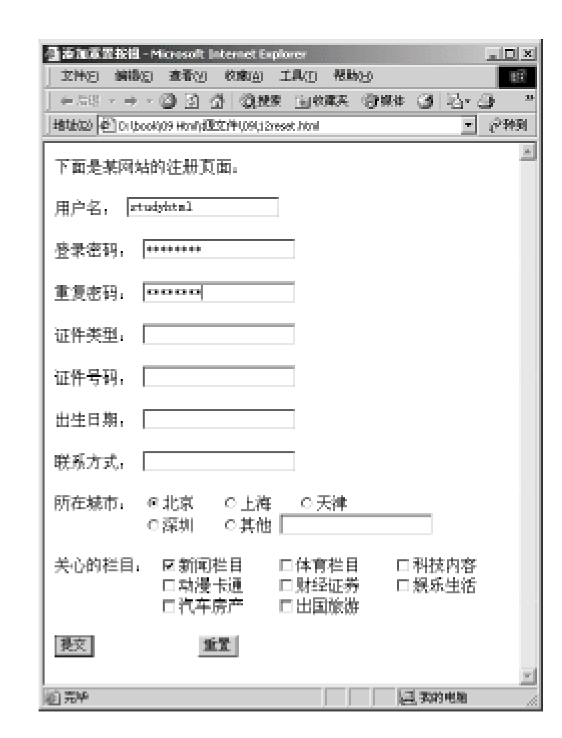


图 9-13　填写部分表单内容

9.3.8　图像域——image

在页面中还有一种控件形式，称为图像域，常用在创建特殊效果的按钮中，因此也常常被称为图像提交按钮。

语法：<input type="image" src="图像地址" name="图像域名称">

说明：在该语法中，图像地址可以是绝对地址或相对地址。

实例代码：

```
<html>
<head>
<title>添加图像提交按钮</title>
</head>
<body>
        下面是关于本网站的调查内容：
        <form action="mailto:abcd@163.com" name="research" method="post">
           <p>登录用户名：
              <input name="username" type="text" size=20>
           </p>
           <p>登录密码：
              <input name="password" type="password" size=20>
           </p>
           <p>重复密码：
              <input name="password2" type="password" size=20>
         </p>
        <hr>
           <p>关心的内容：
               <input name="content" type="checkbox" value="checkbox" checked>新闻栏目   
               <input name="content" type="checkbox" value="checkbox">体育栏目   
               <input name="content2" type="checkbox" value="checkbox">科技内容   
               <input name="content3" type="checkbox" value="checkbox">动漫卡通<br>
                           
               <input name="content4" type="checkbox" value="checkbox">财经证券   
               <input name="content5" type="checkbox" value="checkbox">娱乐生活   
               <input name="content6" type="checkbox" value="checkbox">汽车房产   
               <inputname="content7"  type="checkbox"
value="checkbox">出国旅游</p>
           <p>
              <input type="image" src="pic01.jpg" name="image1">
           </p>
        </form>
</body>
</html>
```

运行这段代码，效果如图 9-14 所示。

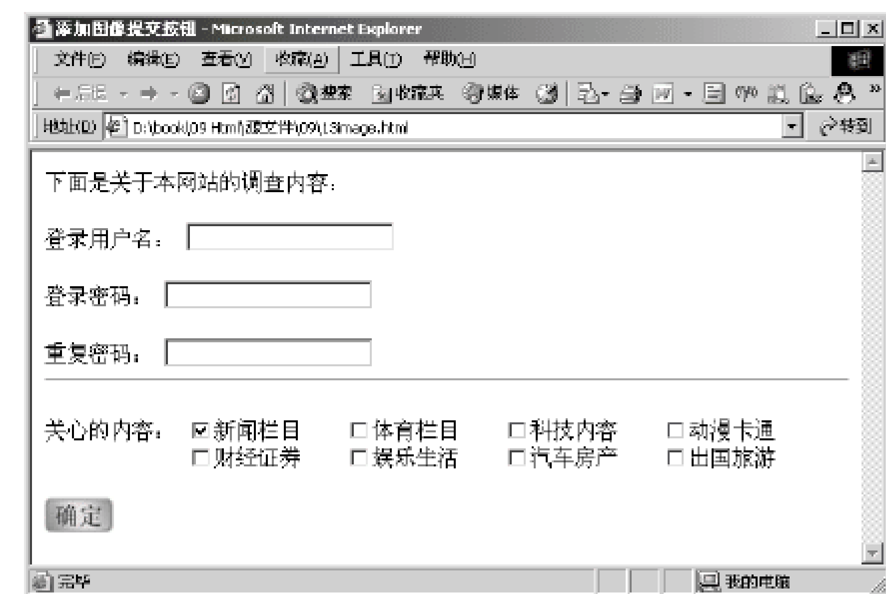


图 9-14　运行代码的效果

9.3.9　隐藏域——hidden

表单中的隐藏域主要用来传递一些参数，而这些参数不需要在页面中显示。当浏览者提交表单时，隐藏域的内容会一起提交给处理程序。

语法：<input type="hidden" name="隐藏域名称" value="提交的值">

实例代码：

```
<html>
<head>
<title>在表单中添加隐藏域</title>
</head>
<body>
    下面是几种不同属性的文字字段：
    <form name="example" action="deal.asp" method="post">
        <!--添加一个长度为 15 的文本框-->
        姓名：<input type="text" name="username" size=15>
        <br>
        <!--添加一个长度为 15，但是最长字符只有 2 的文本框-->
        年龄：<input type="text" name="age" size=15 maxlength=2>
        <br>
        <!--添加一个长度为 15，但最多可输入 30 个字符，默认显示“http://”的文本框-->
        个人主页：<input type="text" name="privateweb" size=15 maxlength=30 value="http://">
        <!--添加隐藏内容-->
        <input type="hidden" name="page_id" value="example ">
    </form>
</body>
</html>
```

运行这段代码，隐藏域的内容并不能显示在页面中，但是在提交表单时，其名称 page_id 和取值 example 将会同时传递给处理程序。

9.3.10　文件域——file

文件域在上传文件时常常用到，它用于查找硬盘中的文件路径，然后通过表单将选中的文件上传。在设置电子邮件的附件、上传头像、发送文件时常常会看到这一控件。

语法：<input type="file" name="文件域的名称">

实例代码：

```
<html>
<head>
<title>添加文件域</title>
</head>
<body>
    下面是某网站的注册页面：
    <form action="mailto:abcd@163.com" name="research" method="post">
```

```
        <p>用户名：
          <input name="username" type="text" size=20>
        </p>
        <p>登录密码：
          <input name="password" type="password" size=20>
        </p>
        <p>重复密码：
          <input name="password2" type="password" size=20>
         </p>
        <p>证件类型：
          <input name="papertype" type="text" size=20>
        </p>
        <p>证件号码：
          <input name="papernum" type="text" size=20 maxlength=35>
        </p>
        <p>联系方式：
          <input name="touch" type="text" size=20 maxlength=50>
        </p>
        <p>所在城市：
            <input name="city" type="radio" value="radiobutton" checked>北京  
            <input name="city" type="radio" value="radiobutton">上海  
            <input name="city" type="radio" value="radiobutton">天津  
            <input name="city" type="radio" value="radiobutton">深圳  
            <input name="city" type="radio" value="radiobutton">其他 <input name="city" type="text">
        </p>
        <p>
        <!--在这里添加了一个文件域-->
        请上传你的头像图片<input type="file" name="picture">
        </p>
        <p>关心的栏目：
            <input name="content" type="checkbox" value="checkbox" checked>新闻栏目   
            <input name="content" type="checkbox" value="checkbox">体育栏目   
            <input name="content2" type="checkbox" value="checkbox">科技内容   
            <input name="content3" type="checkbox" value="checkbox">动漫卡通<br>
                        
            <input name="content4" type="checkbox" value="checkbox">财经证券   
            <input name="content5" type="checkbox" value="checkbox">娱乐生活   
            <input name="content6" type="checkbox" value="checkbox">汽车房产   
            <input name="content7" type="checkbox" value="checkbox">出国旅游</p>
        <p>
          <input type="submit" name="Submit" value="提交">
                     
          <input type="reset" name="reset" value="重置">
        </p>
      </form>
</body>
</html>
```

运行这段代码，可以看到页面中添加了一个“浏览...”按钮，单击这一按钮会打开“选择文件”

对话框，如图 9-15 所示。



图 9-15　添加文件域

9.4　菜单列表类的控件

菜单列表类的控件主要用来进行选择给定答案中的一种，这类选择往往答案比较多，使用单选按钮比较浪费空间。可以说，菜单列表类的控件主要是为了节省页面空间而设计的。菜单和列表都是通过<select>和<option>标记来实现的。

9.4.1　下拉菜单

下拉菜单是一种最节省页面空间的选择方式，因为在正常状态下只显示一个选项，单击按钮打开菜单后才会看到全部的选项。

语法：

```
<select name="下拉菜单的名称">
        <option value="选项值" selected>选项显示内容
    <option value="选项值">选项显示内容
    ……
</select>
```

说明：在该语法中，选项值是提交表单时的值，而选项显示的内容才是真正在页面中显示的选项。selected 表示该选项默认情况下是选中的，一个下拉菜单中只能有一项默认被选中。

实例代码：

```
<html>
<head>
<title>添加下拉菜单</title>
</head>
<body>
```

```
下面是某网站的注册页面：
<form action="mailto:abcd@163.com" name="research" method="post">
  <p>用户名：
    <input name="username" type="text" size=20>
  </p>
  <p>登录密码：
    <input name="password" type="password" size=20>
  <p>
 <p> 重复密码：
    <input name="password2" type="password" size=20>
 </p>
  <p>证件类型：
    <select name="cardtype">
        <option value="id_card" selected>身份证
        <option value="stu_card">学生证
        <option value="drive_card">驾驶证
        <option value="other_card">其他证件
    </select>
  </p>
  <p>证件号码：
    <input name="cardrnum" type="text" size=20 maxlength=35>
  </p>
  <p>联系方式：
    <input name="touch" type="text" size=20 maxlength=50>
  </p>
  <p>关心的栏目：
     <input name="content" type="checkbox" value="checkbox" checked>新闻栏目   
     <input name="content" type="checkbox" value="checkbox">体育栏目   
     <input name="content2" type="checkbox" value="checkbox">科技内容   
     <input name="content3" type="checkbox" value="checkbox">动漫卡通<br>
                 
     <input name="content4" type="checkbox" value="checkbox">财经证券   
     <input name="content5" type="checkbox" value="checkbox">娱乐生活   
     <input name="content6" type="checkbox" value="checkbox">汽车房产   
     <input name="content7" type="checkbox" value="checkbox">出国旅游</p>
  <p>
    <input type="submit" name="Submit" value="提交">
               
    <input type="reset" name="reset" value="重置">
  </p>
</form>
</body>
</html>
```

运行这段代码，可以看到页面中添加了包含 4 个选项的下拉菜单，其中“身份证”选项被设置为默认，如图 9-16 所示。

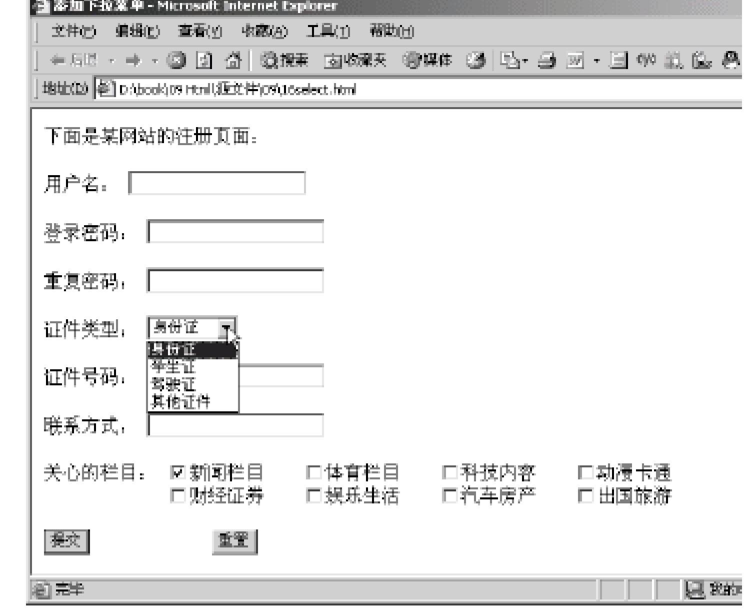


图 9-16　添加下拉菜单

9.4.2　列表项

列表项的设置方法与下拉菜单类似。不同的是，列表项在页面中可以显示出几条信息，一旦超出这个信息数量，在列表右侧会出现滚动条，拖动滚动条可以看到所有的选项。

语法：

```
<select name="列表项名称" size="显示的列表项数" multiple>
    <option value="选项值" selected>选项显示内容
    <option value="选项值">选项显示内容
    ……
</select>
```

说明：在该语法中，size 设定页面中的最多列表项数，当超过这个值时会出现滚动条。multiple 表示这一列表可以进行多项选择。选项值是提交表单时的值，而选项显示内容才是真正在页面中显示的选项。

实例代码：

```
<html>
<head>
<title>添加列表项</title>
</head>
<body>
    下面是某网站的注册页面：
    <form action="mailto:abcd@163.com" name="research" method="post">
      <p>用户名：
        <input name="username" type="text" size=20>
      </p>
      <p>登录密码：
        <input name="password" type="password" size=20>
      </p>
      <p>重复密码：
        <input name="password2" type="password" size=20>
      </p>
      <p>证件类型：
        <select name="cardtype">
          <option value="id_card" selected>身份证
          <option value="stu_card">学生证
          <option value="drive_card">驾驶证
          <option value="other_card">其他证件
        </select>
      </p>
      <p>证件号码：
        <input name="cardrnum" type="text" size=20 maxlength=35>
      </p>
      <p>联系方式：
        <input name="touch" type="text" size=20 maxlength=50>
```

```
        </p>
        <p>关心的栏目：
           <select name="content" size=5 multiple>
              <option value="M1" selected>体育栏目
              <option value="M2">科技内容
              <option value="M3">新闻栏目
              <option value="M4">卡通动漫
              <option value="M5" selected>财经证券
              <option value="M6">娱乐生活
              <option value="M7">汽车房产
              <option value="M8">出国旅游
           </select>
        </p>
        <p>
          <input type="submit" name="Submit" value="提交">
                     
          <input type="reset" name="reset" value="重置">
        </p>
      </form>
</body>
</html>
```

运行这段代码，可以看到页面上添加了一个列表选项，其中显示的选项个数为 5，并可以进行多项选择，其中“体育栏目”和“财经证券”两项是默认被选中的栏目，如图 9-17 所示。

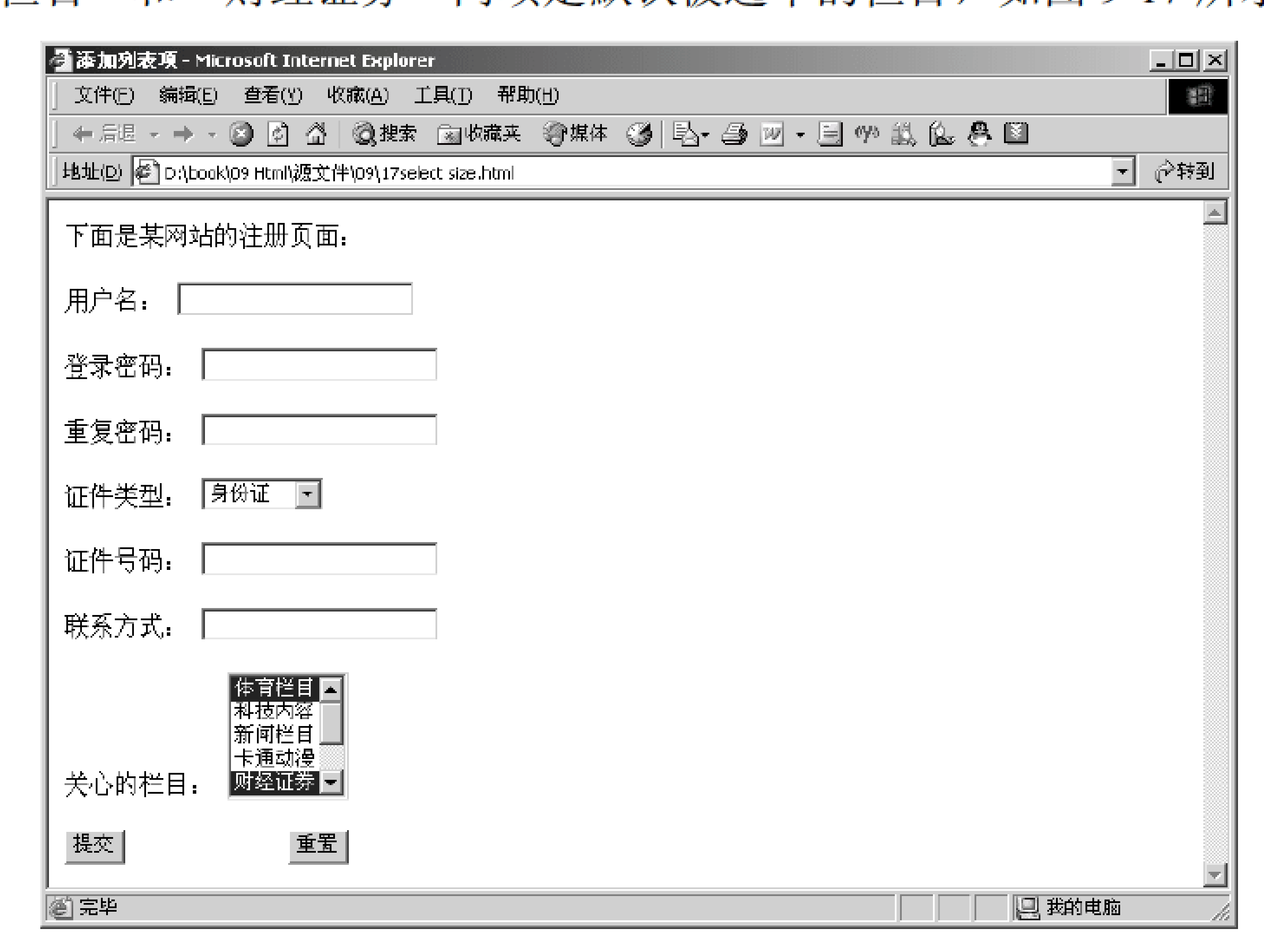


图 9-17　添加列表项

9.5　文本域标记——textarea

除了以上讲解的两大类控件外，还有一种特殊定义的文本样式，称为文字域或文本域。它与文字字段的区别在于可以添加多行文字，从而可以输入更多的文本。这类控件在一些留言板中最为常见。

语法：

```
<textarea name="文本域名称" value="文本域默认值" rows=行数 cols=列数>
</textarea>
```

说明：在该语法中，rows 是指文本域的行数，也就是高度值，当文本内容超出这一范围时会出现滚动条；cols 设置文本域的列数，也就是其宽度。

实例代码：

```
<html>
<head>
<title>添加文本域</title>
</head>
<body>
    下面是某网站的留言页面：
    <form action="mailto:abcd@163.com" name="research" method="post">
      <p>用户名：
        <input name="username" type="text" size=20>
      </p>
      <p>密  码：
        <input name="password" type="password" size=20>
      </p>
      <p>留言：<br>
        <textarea name="word" rows=5 cols=70>
        </textarea>
      </p>
      <p>
        <input type="submit" name="Submit" value="提交">
              
        <input type="reset" name="reset" value="重置">
      </p>
    </form>
</body>
</html>
```

运行代码，可以看到页面上添加了一个行数为 5、列数为 70 的文本域，如图 9-18 所示。

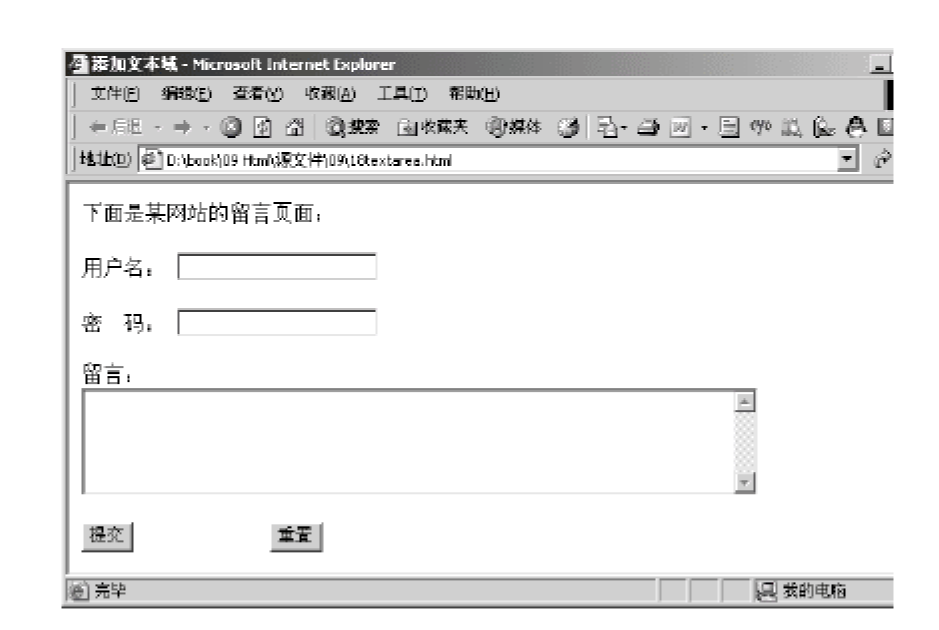


图 9-18　添加文本域的效果

9.6　id 标记

在 HTML 的表单元素中，还有一个 id 标记。这一标记是一个较为特殊的标记，它主要用于标示一个惟一的元素。这个元素可以是文字字段，可以是密码域，也可以是其他的表单元素，甚至也可以定义一幅图像、一个表格。但是在实际应用中，表单是使用 id 标记最多的一类元素。

基本语法：<id="元素的标识名">

说明：在 HTML 中，由于 id 用来标识页面的惟一元素，因此在定义标识名时最好要根据其含义进行命名。

实例代码：

```
<html>
<head>
<title>添加文本域</title>
</head>
<body>
    下面是某网站的留言页面：
    <form action="mailto:abcd@163.com" name="research" method="post">
      <p>用户名：
        <input name="username" type="text" size=20 id="username">
      </p>
      <p>密  码：
        <input name="password" type="password" size=20>
      </p>
      <p>留言：<br>
        <textarea name="word" rows=5 cols=70>
        </textarea>
      </p>
      <p>
        <input type="submit" name="Submit" value="提交">
                   
        <input type="reset" name="reset" value="重置">
      </p>
    </form>
</body>
</html>
```

在该实例中，定义了用户名的文字字段 id 为 username。而在运行程序时，页面中并不显示该 id，只是在将信息传送到服务器时会同时被提交。



第10章 框架结构

- 窗口框架简介
- 设置框架集的基本属性
- 设置窗口属性
- 浮动框架
- 框架与链接

框架的最主要功能是用来“分割”页面窗口，使每个“小窗口”能显示不同的 HTML 文件，这样的页面结构就称为框架结构的页面，而这些“小窗口”就被称为框架的“窗口”。可以说框架就是将网页画面分成几个窗口，同时取得多个 URL。本章将详细讲解各种框架标记的使用。

10.1　窗口框架简介

10.1.1　什么是框架

如果页面可以分为几个部分，各个部分之间是相互独立的页面，却又互相有关联，用户在浏览这种页面时，当对其中某一部分进行操作，如浏览、下载时，其他页面会保持不变，这样的页面就被称为框架结构的页面，也称为多窗口页面。

实际上框架对象本身也是一类窗口，它继承了窗口对象的所有特征，并拥有所有的属性和方法。

使用框架最主要的目的就是创建链接的结构，最常见的框架结构就是将网站的导航条作为一个单独的框架窗口，当用户查看具体的内容时，导航条窗口保持不变，这就为用户的浏览提供了方便。

10.1.2　框架的基本结构

框架主要包含两个部分，一个是框架集，另一个就是具体的框架文件。

框架集就是用来定义这一 HTML 文件为框架模式，并设定视窗如何分割的文件。通俗一点地说，框架集就是存放框架结构的文件，也是访问框架文件的入口文件。如果网页由左右两个框架组成，那么除了左右两个网页文件之外，还有一个总的框架集文件。

框架是页面中定义的每一个显示区域，也可以说一个窗口就是一个框架。框架页面中最基本的内容就是框架集文件，它是整个框架页面的导航文件，其基本语法如下；

```
<html>
<head>
<title>框架页面的标题</title>
</head>
<frameset>
    <frame>
    <frame>
    ……
</frameset>
</html>
```

从上面的语法结构中可以看到，在使用框架的页面中，<body>主体标记被框架标记<frameset>所代替。而对于框架页面中包含的每一个框架，都是通过<frame>标记来定义的。

10.2　设置框架集的基本属性

框架页面的结构也是在框架集文件中定义的，一般情况下，根据框架的分割方式来分类，主要包

含 3 种框架结构，分别是：

- 水平分割窗口。
- 垂直分割窗口。
- 嵌套分割窗口。

下面对框架集的结构和一些基本属性进行详细的说明。

10.2.1　水平分割窗口——rows

水平分割窗口是将页面沿水平方向切割，也就是将页面分成上下排列的多个窗口。

语法：

```
<frameset rows="框架窗口的高度,框架窗口的高度,……">
    <frame>
    <frame>
    ……
</frameset>
```

说明：在该语法中，rows 中可以取多个值，每个值表示一个框架窗口的水平宽度，它的单位可以是像素，也可以是占浏览器的百分比。但是要注意的是，一般设定了几个 rows 的值，就需要有几个框架，即需要有相应数量的<frame>参数。

实例代码：

```
<html>
<head>
<title>水平分割窗口的效果</title>
</head>
<frameset rows="30%,70%">
    <frame>
    <frame>
</frameset>
</html>
```

运行代码，可以看到页面被分割成上下两个窗口，效果如图 10-1 所示。当浏览器大小变化时，框架也会随之等比例缩放。

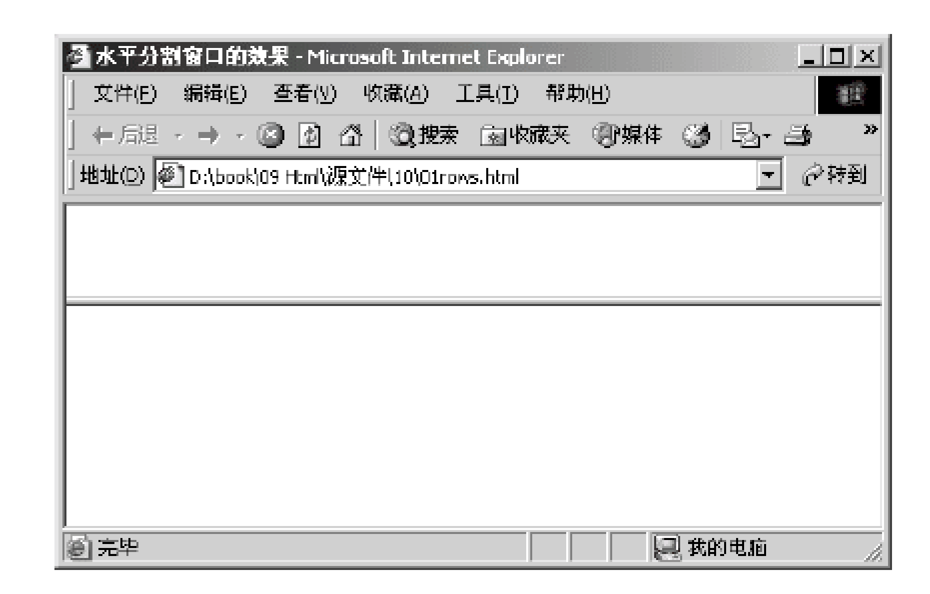


图 10-1　水平分割窗口

10.2.2　垂直分割窗口——cols

垂直分割窗口就是将页面沿垂直方向分割成多个窗口，也就是将页面分成左右排列的多个窗口。

语法：

```
<frameset cols="框架窗口的宽度,框架窗口的宽度,……">
    <frame>
    <frame>
    ……
</frameset>
```

说明：在该语法中，cols 中可以取多个值，每个值表示一个框架窗口的水平宽度，它的单位可以是像素，也可以是占浏览器的百分比。与水平分割窗口相同，一般设定了几个 cols 的值，就需要有几个框架，也就是有几个<frame>参数。

实例代码：

```
<html>
<head>
<title>垂直分割窗口的效果</title>
</head>
<frameset cols="20%,55%,25%">
    <frame>
    <frame>
    <frame>
</frameset>
</html>
```

运行代码，可以看到页面被分割成 3 个窗口，其中左右两个宽度相同，效果如图 10-2 所示。当浏览器大小变化时，框架也会随之等比例缩放。

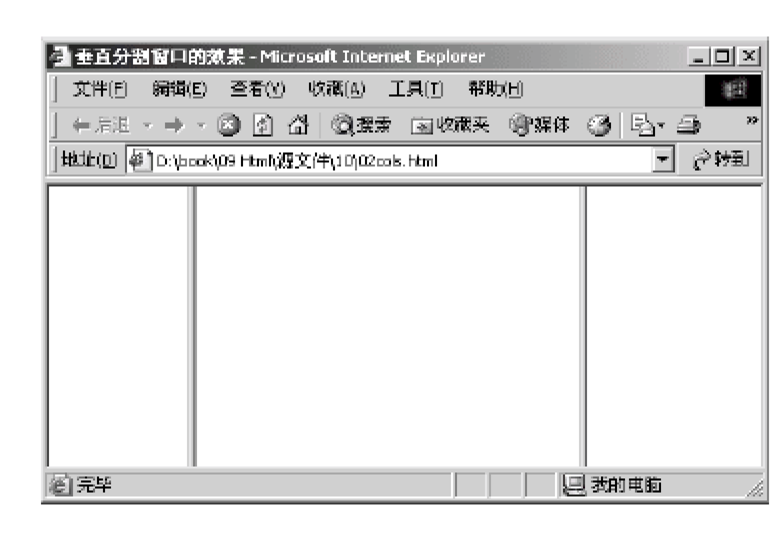


图 10-2　垂直分割窗口

10.2.3　嵌套分割窗口

嵌套分割窗口就是在一个页面中，既有水平分割的框架，又有垂直分割的框架。

语法：

```
<frameset rows="框架窗口的高度,框架窗口的高度,……">
    <frame>
      <frameset cols="框架窗口的宽度,框架窗口的宽度,……">
        <frame>
        <frame>
        ……
      </frameset>
```

```
        <frame>
        ……
    </frameset>
```

当然，也可以先进行垂直分割，再进行水平分割。其语法如下：

```
    <frameset cols="框架窗口的宽度,框架窗口的宽度,……">
        <frame>
          <frameset rows="框架窗口的高度,框架窗口的高度,……">
            <frame>
            <frame>
            ……
          </frameset>
        <frame>
        ……
    </frameset>
```

这两种结构的原理与注意事项和另外两种结构相同，主要是需要注意窗口大小的设置与窗口个数的统一。

实例代码：

```
<html>
<head>
<title>嵌套分割窗口的效果</title>
</head>
<frameset rows="30%,70%">
        <frame>
          <frameset cols="20%,55%,25%">
             <frame>
             <frame>
             <frame>
          </frameset>
</frameset>
</html>
```

由代码可以看出，首先将页面进行水平分割成上下两个窗口，接着下面的框架又被垂直分割成 3 个窗口。因此下面的框架标记<frame>被框架集标记代替。运行程序，效果如图 10-3 所示。

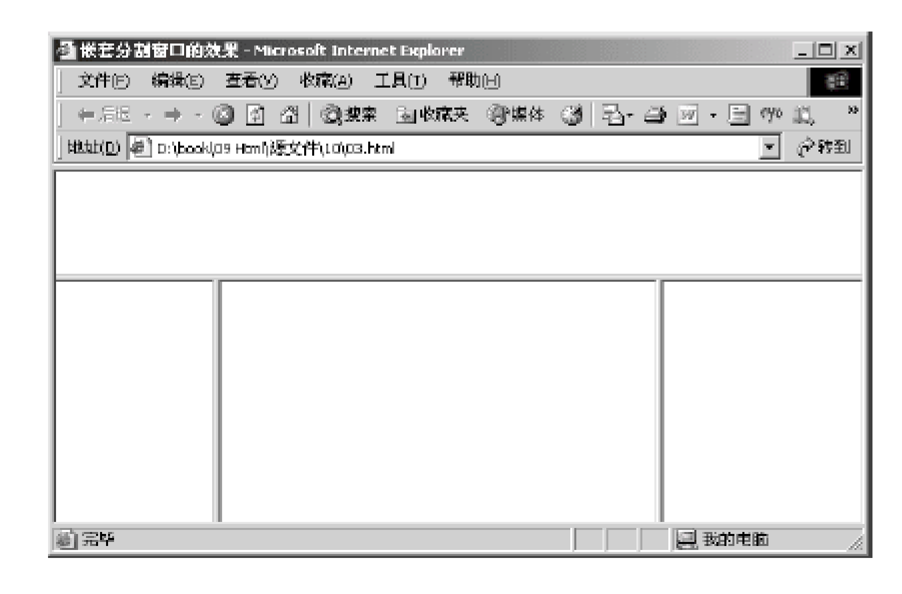


图 10-3　嵌套分割的效果

10.2.4　设置边框——frameborder

由前面的几个实例可以看出，在默认情况下，框架窗口的四周有一条边框线，通过 frameborder 参数可以调整边框线的显示情况。

语法：<frameset frameborder="是否显示">　或<frame frameborder="是否显示">

说明：frameborder 的取值只能为 0 或 1。如果取值为 0，那么边框线将会被隐藏；如果取值为 1，边框线将会显示。在 frameset 中设置将会对整个框架有效，在 frame 中设置则只对当前这个框架有效。

实例代码：

```
<html>
<head>
<title>设置框架窗口的边框显示效果</title>
</head>
<frameset rows="20%,55%,25%">
      <frame frameborder="1">
        <frameset cols="35%,65%" frameborder="0">
            <frame >
            <frame>
        </frameset>
      <frame frameborder="0">
</frameset>
</html>
```

运行这段代码，可以看到页面中的部分边框被隐藏，如图 10-4 所示。当鼠标移动到窗口中间时会变成↔，按下鼠标会看到隐藏的边框，如图 10-5 所示。



图 10-4　隐藏框架的边框

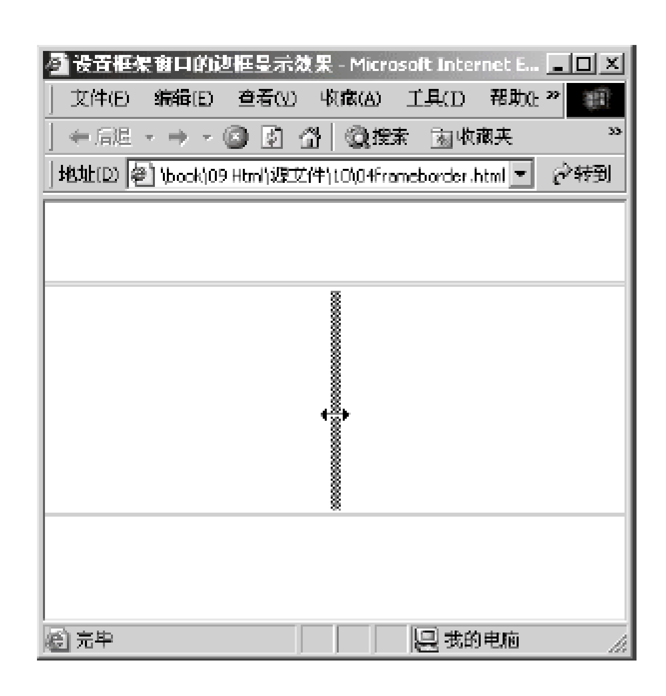


图 10-5　按下鼠标的效果

10.2.5　框架的边框宽度——framespacing

框架的边框宽度在默认情况下是 1 像素，通过参数 framespacing 可以调整其大小。

语法：<frameset framespacing="边框宽度">

说明：边框宽度就是在页面中各个边框之间的线条宽度，以像素为单位。而这一参数只能对框架集使用，对单个框架无效。

实例代码：

```
<html>
<head>
<title>设置框架的边框宽度</title>
</head>
<frameset rows="30%,70%" framespacing="10">
       <frame>
          <frameset cols="20%,55%,25%" framespacing="30">
              <frame>
              <frame>
              <frame>
          </frameset>
</frameset>
</html>
```

运行这段程序，效果如图 10-6 所示。

图 10-6　设置框架的边框宽度

10.2.6　框架的边框颜色——bordercolor

使用参数 bordercolor 可以设置框架集的边框颜色。

语法：<frameset bordercolor="颜色代码">

说明：该参数同样只对整个框架集有效，对于单个框架无效。

实例代码：

```
<html>
<head>
<title>设置框架的边框颜色</title>
</head>
<frameset rows="30%,70%" framespacing="10" bordercolor ="#CC99FF">
       <frame>
          <frameset cols="20%,55%,25%" framespacing="30" bordercolor ="#9900FF">
              <frame>
              <frame>
              <frame>
          </frameset>
</frameset>
</html>
```

运行这段代码，可以看到两个框架集设置了不同的边框颜色，效果如图 10-7 所示。

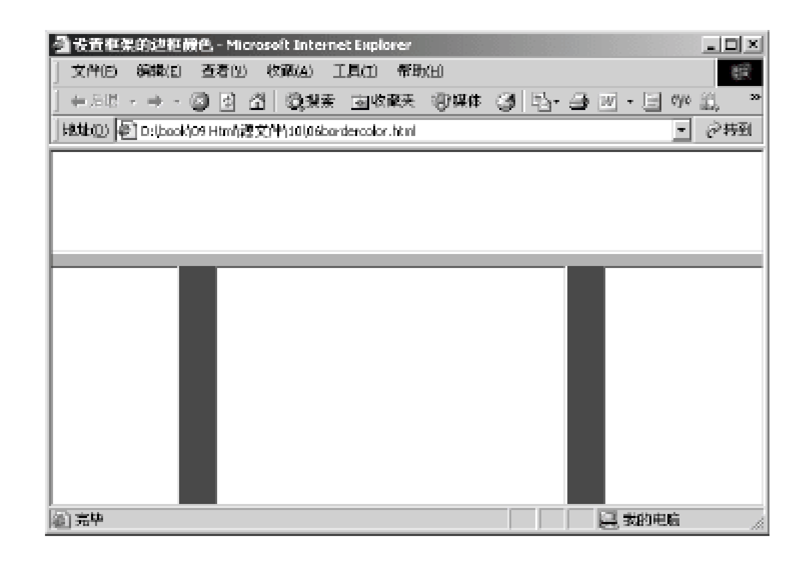


图 10-7　设置边框颜色

10.3　设置窗口属性

在框架结构的页面中，每一个框架窗口都有一个显示页面，这个页面称为框架页面。框架页面的属性设置都在<frame>标记里进行。

10.3.1　页面源文件——src

框架结构中的各个页面都是一个单独的文件，而这些文件是通过 src 参数进行设置的。

语法：<frame src="页面源文件地址">

说明：页面文件是框架页面的具体内容所在，对于没有设置源文件的框架，只是空白页面，是没有任何作用的。页面的源文件可以是正常的 HTML 文件，也可以是一个图片或者其他的文件。

为了说明框架的参数效果，首先设置一个框架页面的内容，代码如下：

```
<html>
<head>
<title>段落的缩进效果</title>
</head>
<body>
    相传，两千五百年前，春秋时期的大音乐家俞伯牙，曾学琴于程廉先生，三年不成。后来他沿着孔子的足迹
登游泰山，
    <blockquote>观东海日出，看云雾变化，</blockquote>
    <blockquote><blockquote>闻松风长啸，听水涛咆哮，</blockquote></blockquote>
    <blockquote><blockquote><blockquote>拜大自然为师，琴艺大有长进，
</blockquote></blockquote></blockquote>
    <blockquote><blockquote><blockquote><blockquote>写出了著名的古琴曲高山和流水。
</blockquote></blockquote></blockquote></blockquote>
</body>
</html>
```

将这一文件命名为 src01.html，将其保存在框架集文件的同一目录下，然后再设置框架集文件的代码如下：

```
<html>
<head>
<title>设置页面源文件</title>
</head>
<frameset rows="30%,70%">
    <frame src="pic01.gif">
    <frame src="src01.html">
</frameset>
</html>
```

运行这段代码，效果如图 10-8 所示。

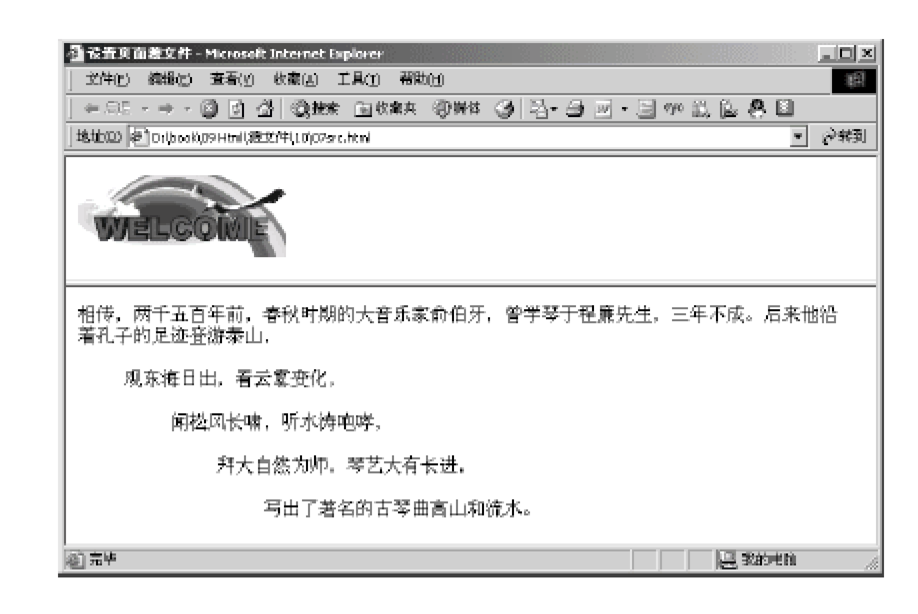


图 10-8　设置页面的源文件

10.3.2　页面名称——name

页面名称是为了便于页面的查找和链接所提供的一个属性。框架的页面名称中不允许包含特殊字符、连字符-、空格等，必须是单个的单词或者字母组合。

语法：<frame src="页面文件地址" name="页面名称">

实例代码：

```
<html>
<head>
<title>设置页面名称</title>
</head>
<frameset rows="30%,70%">
    <frame src="pic01.gif" name="pic">
    <frame src="src01.html" name="text">
</frameset>
</html>
```

这段代码分别为上下两个框架页面命名为 pic 和 text，运行代码时并不会显示框架名，因此不会影响框架的显示效果。

10.3.3　调整窗口的尺寸——noresize

由图 10-5 可以看出，当用鼠标拖动框架的边框时，框架窗口的尺寸就会随之变化。如果希望框架窗口的大小保持不变，可以设置 noresize 参数。

语法：<frame src="页面文件地址" noresize>

说明：在 frame 标记中添加 noresize 参数，就表示框架窗口的尺寸不能改变。

实例代码：

```
<html>
<head>
<title>禁止改变框架窗口的尺寸</title>
</head>
<frameset rows="30%,70%">
    <frame src="pic01.gif" name="pic" noresize>
    <frame src="src01.html" name="text">
</frameset>
</html>
```

运行这段代码后，将鼠标放置在框架边框上时，光标不会变为双箭头形状 ↕，也无法拖动框架的边框，如图 10-9 所示。

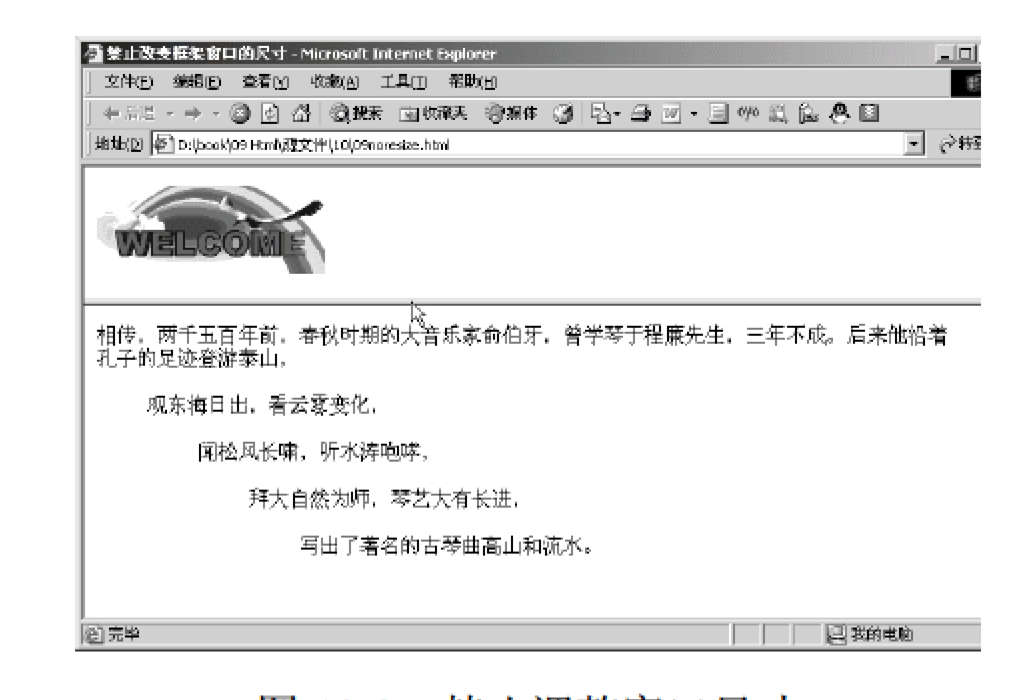


图 10-9　禁止调整窗口尺寸

10.3.4　边框与页面内容的水平边距——marginwidth

框架页面与 HTML 中的表格一样，也可以设置边框与页面内容的水平边距。

语法：<frame src="页面文件地址" marginwidth="水平边距">

说明：水平边距用于设置页面的左右边缘与框架边框的距离，单位是像素。

首先创建一个 src02.html 页面，代码如下：

```
<html>
<head>
<title>设置水平边距</title>
</head>
<body>
	<br>
	<h3><font color="#009933">美丽的江南小镇</font></h3>
	<hr>
	<font color="#009966"><br>
	穿镇而过的狭窄河道，一座座雕刻精致的石桥，傍河而筑的民居，民居楼板底下就是水，石阶的埠头从楼板下一级级伸出来，女人正在埠头上浣洗，而离他们只有几尺远的乌蓬船上正升起一缕白白的炊烟，炊烟穿过桥洞飘到对岸，对岸河边有又低又宽的石栏，可坐可躺，几位老人满脸宁静地坐在那里看着过往船只。比之于沈从文笔下的湘西河边由吊脚楼组成的小镇，江南小镇少了那种浑朴奇险，多了一点畅达平稳。它们的前边没有险滩，后边没有荒漠，因此虽然幽僻却谈不上什么气势；它们大多很有一些年代了，但始终比较滋润的生活方式并没有让它们保留下多少废墟和遗迹，因此也听不出多少历史的浩叹；它们当然有过升沉荣辱，但实在也未曾摆出过太堂皇的场面，因此也不容易产生类似于朱雀桥、乌衣巷的沧桑之慨。总之，它们的历史路程和现实风貌都显得平实而耐久，狭窄而悠久，就像经纬着它们的条条石板街道。
	</font>
</body>
</html>
```

创建框架页面集代码如下：

```
<html>
<head>
<title>设置水平边距</title>
</head>
<frameset cols="35%,65%">
	<frame src="pic02.jpg" name="pic" >
	<frame src="src02.html" name="text" marginwidth="70">
</frameset>
</html>
```

运行这段代码，可以看到右侧的框架页面中，文本内容与框架的边框之间设置了很大的空间，从而显得宽松很多，如图 10-10 所示。



图 10-10　设置水平边距

10.3.5　边框与页面内容的垂直边距——marginheight

与水平边距类似，通过 marginheight 参数可以设置边框与页面的垂直边距。

语法：<frame src="页面文件地址" marginheight="垂直边距">

说明：垂直边距用来设置页面的上下边缘与框架边框的距离，单位是像素。

实例代码：

```
<html>
<head>
<title>设置垂直边距</title>
</head>
<frameset cols="35%,65%">
      <frame src="pic02.jpg" name="pic" >
      <frame src="src02.html" name="text" marginwidth="70" marginheight="90">
</frameset>
</html>
```

运行这段代码，可以看到右侧的框架页面中，文本内容与框架的上下边框之间设置了很大的空间，如图 10-11 所示。

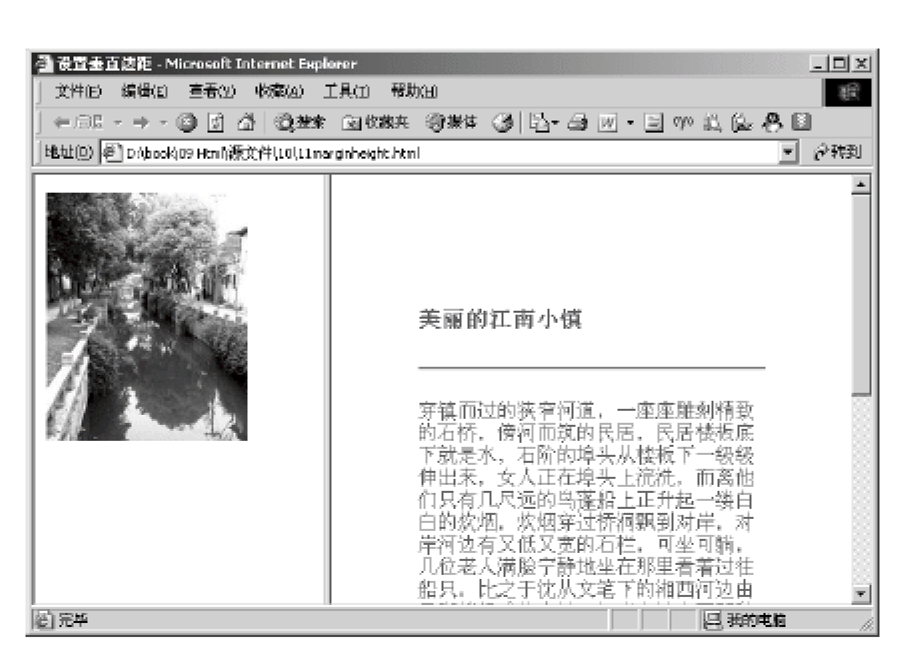


图 10-11　设置垂直边距的效果

10.3.6　设置框架滚动条显示——scrolling

在默认情况下，当页面的内容超出浏览器窗口的大小时，页面会自动添加滚动条以方便用户浏览。但是有时用户希望不显示滚动条，这可以通过设置 scrolling 参数来实现。

语法：<frame src="页面源文件" scrolling="是否显示滚动条">

说明：scrolling 参数在这里只能取 Yes、No 或 Auto 三个值中的一个。其中，Yes 表示一直显示滚动条，而 No 则表示无论什么情况都不显示滚动条；Auto 是系统的默认值，它是根据具体内容来调整的，也就是说当页面长度超出浏览器窗口的范围时就会自动显示滚动条。

实例代码：

```
<html>
<head>
```

```
<title>设置垂直边距</title>
</head>
<frameset cols="35%,65%">
      <frame src="pic02.jpg" name="pic" scrolling="Yes">
      <frame src="src02.html" name="text" marginwidth="70" marginheight="90" scrolling="No">
</frameset>
</html>
```

运行这段代码，可以看到如图 10-12 所示的效果。与图 10-11 相对比，可以看出设置 scrolling 为 Yes 时，即使页面内容远远小于浏览器的大小，也会显示滚动条；而当 scrolling 设置为 No 时，即使内容超出了浏览器的范围，也没有显示滚动条。

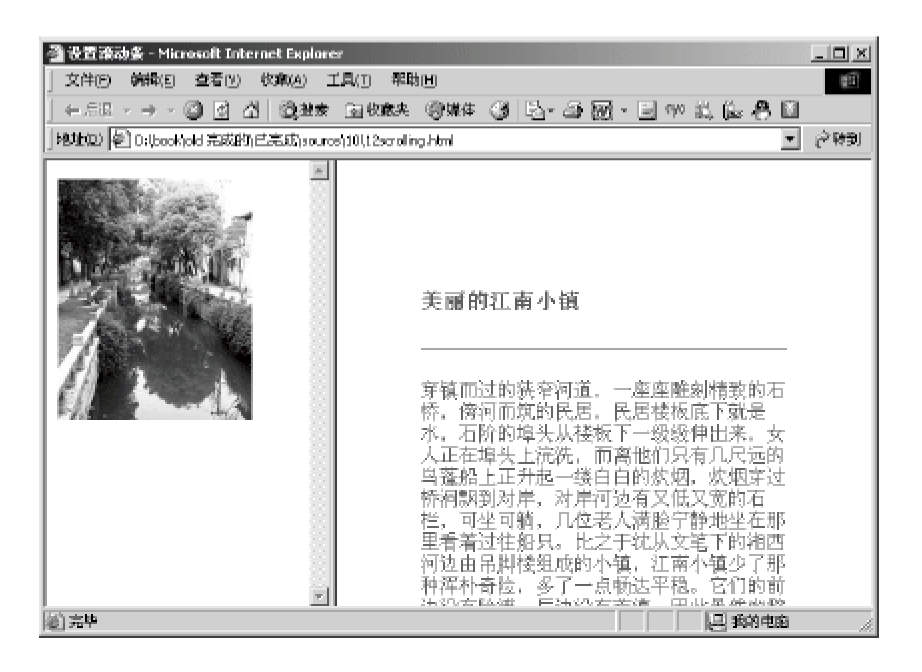


图 10-12　设置滚动条的效果

10.3.7　不支持框架标记——noframes

对于一些低版本的浏览器来说，如果不支持框架结构就无法打开框架页面，但这样还不够，因为它并不清楚为什么打不开页面。通过 noframes 参数可以设置一些内容（包括文字或图片）来告诉浏览者其浏览器无法打开框架页面。

语法：

```
<html>
<head>
<title>框架页面的标题</title>
</head>
<frameset>
      <frame>
      <frame>
      ……
      <noframe>
        ……
      </noframe>
</frameset>
</html>
```

说明：在该语法中，<noframe>和</noframe>标记之间的部分就是在不支持框架的浏览器中所要显示的内容。它可以与<body>标记一样添加内容。

实例代码：

```
<html>
<head>
<title>设置水平边距</title>
</head>
<frameset cols="35%,65%">
     <frame src="pic02.jpg" name="pic" >
     <frame src="src02.html" name="text">
     <noframe>
          <center>
            <p>对不起，您的浏览器不支持框架页面，因此无法显示此
页面！</p>
            <img src="pic03.jpg">
          </center>
     </noframe>
</frameset>
```

运行这段代码后，当用户的浏览器不支持框架结构时，就是直接显示<noframe>标记内如图 10-13 所示的页面内容。



图 10-13　显示<noframe>标记内的内容

10.4　浮 动 框 架

浮动框架是一种较为特殊的框架，它是在浏览器窗口中嵌套子窗口，也就是整个页面并不是框架页面，但是却包含一个框架窗口，在框架窗口内显示相应的页面内容。

语法：

```
<iframe src="页面源文件">
</iframe>
```

说明：与普通框架结构类似，浮动框架也可以设置很多参数，见表 10-1。

表10-1　浮动框架的参数设置

浮动框架可以设置的参数	参数的含义
src	浮动框架页面的源文件地址
width	浮动框架在页面中显示的宽度
height	浮动框架在页面中显示的高度
align	浮动框架页面在浏览器中的对齐方式，可以为左对齐、右对齐或居中对齐
name	设定框架页面的名称
marginwidth	设置页面与边框的水平间距
marginheight	设置页面与边框的垂直间距

续表

浮动框架可以设置的参数	参数的含义
scrolling	设定浮动框架页面内是否显示滚动条
frameborder	设定浮动框架的边框

从表 10-1 中可以看出，很多普通框架页面的属性在浮动框架页面中同样适用，例如 name、scrolling 等，而在普通框架中只对框架集有效的一些参数在这里同样可以设置，如 frameborder 等，此外浮动框架页面还可以设置大小和对齐方式。而对于浮动框架来说，框架边框的宽度和颜色是无法设置的，即与普通框架相比，浮动框架中不包含 framespacing 和 bordercolor 参数。

本节将主要通过设置框架页面的参数和不设置进行对比，从而清晰地说明各个参数的功能。为了达到这一目的，需要进行如下的准备工作：创建两个普通的 HTML 页面，其中一个作为浮动框架的载体，命名为 float.html；另一个作为浮动框架页面的源文件，命名为 source.html。

float.html 文件的源代码如下：

```
<html>
<head>
<title>未添加浮动框架的页面</title>
</head>
<body>
	<font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
	<hr size=2>
</body>
</html>
```

source.html 文件的源代码如下：

```
<html>
<head>
<title>浮动框架的页面内容</title>
</head>
<body>
	<font size="5" color="#000099">经典影片：魂断蓝桥</font><br><br>
	<image src="pic04.jpg" align="left">
	《魂断蓝桥》作为电影史上三大凄美不朽爱情影片之一，是一部荡气回肠的爱情经典之作，内容虽有些传奇化，但文艺气息浓厚，具有甚高的催泪效果。<br>
	《魂断蓝桥》（又名《滑铁卢桥》）是一部风靡全球近半个世纪的美国爱情故事片，也是西方电影在东方获得成功的经典。<br>
	<hr>
	剧情介绍：<br>
	一次偶然的机会，芭蕾舞女演员玛亚在滑铁卢桥邂逅了高级军官罗伊。由于战争的原因，两人决定马上结婚，但就在婚礼即将举行的前一天晚上，罗伊接到命令，部队当晚开拔。玛亚无意中看到了罗伊的名字在阵亡名单中。此时罗伊的母亲来看她，尽管这位贵夫人非常和蔼可亲，但此时的玛亚已情绪混乱、言语无礼、不知所云。为了维持生活，玛亚和她的好友都沦为街头应招女郎。罗伊并没有死，他回来了。玛亚的遭遇使她无法面对与罗伊的婚姻及罗伊家族的显赫地位。她来到滑铁卢桥，毫无畏惧地向一辆辆飞驰的军车走去。
</body>
</html>
```

完成了两个页面文件之后，下面开始设置浮动框架页面的内容。

10.4.1　必需参数：页面源文件——src

浮动框架中最基本的参数就是 src，它用来设置浮动框架页面的源文件地址，也是浮动框架页面必需的参数。因为只有设置了源文件的内容，浮动框架才有意义。

语法：<iframe src="页面源文件">

下面将文件 source.html 作为浮动框架页面的源文件插入到 HTML 文件 float.html 中，实例代码如下：

```
<html>
<head>
<title>添加浮动框架的页面</title>
</head>
<body>
    <font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
    <hr size=2>
    <iframe src="source.html">
    </iframe>
</body>
</html>
```

运行程序，效果如图 10-14 所示。

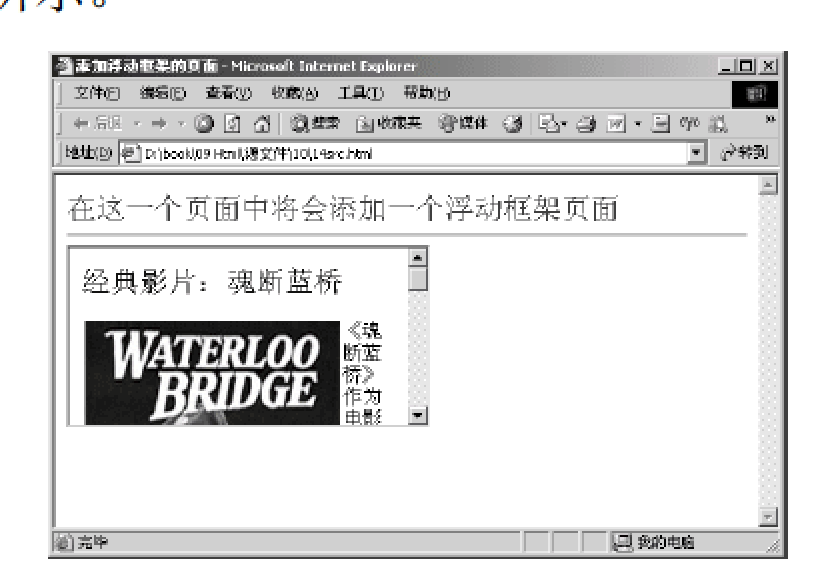


图 10-14　在页面中添加浮动框架页面

下面介绍在浮动框架页面中可以设置的一些框架页面不能设置的参数，称为浮动框架特有参数，主要包括框架的页面大小以及对齐方式。

10.4.2　特有属性：大小——width 和 height

在普通框架结构中，由于框架就是整个浏览器窗口，因此不需要设置其大小。但是在浮动框架中，框架是插入到普通 HTML 页面中的，所以可以调整整个框架的大小。

语法：<iframe src="浮动框架页面源文件" width="页面宽度" height="页面高度">

说明：在该语法中，页面的宽度和高度值都以像素为单位。

下面在 float.html 文件的基础上设置浮动框架页面的大小，实例代码如下：

```
<html>
<head>
```

```
<title>设置框架页面的大小</title>
</head>
<body>
    <font size="5" color="#CC0000">在这一个页面中将会添加一个浮动
框架页面</font>
    <hr size=2>
    <iframe src="source.html" width="550" height="310">
    </iframe>
</body>
</html>
```

运行这段代码，效果如图 10-15 所示，与图 10-14 相对比，可以看到页面中的框架大小被调整得更加协调。

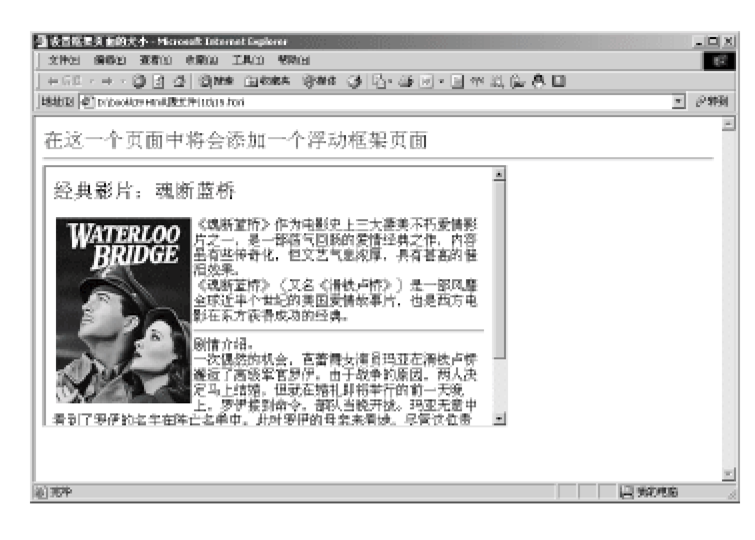


图 10-15　调整浮动框架页面的大小

10.4.3　特有属性：对齐方式——align

除了设置框架大小的参数之外，设置框架的对齐方式也是浮动框架页面所特有的参数。

语法：<iframe src="浮动框架页面源文件" align="对齐方式">

说明：在该语法中，对齐方式用来设置浮动框架页面相对于浏览器窗口的水平位置。它可以取的值包括左对齐 left、右对齐 right 和居中对齐 center。

下面在 float.html 文件的基础上设置浮动框架页面的对齐方式，实例代码如下：

```
<html>
<head>
<title>设置对齐方式</title>
</head>
<body>
    <font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
    <hr size=2>
    <iframe src="source.html" width="550" height="310" align="center">
    </iframe>
</body>
</html>
```

运行这段代码，效果如图 10-16 所示，与图 10-15 相对比，可以看到页面中的框架位置被调整，由原来的左对齐变成了居中对齐。

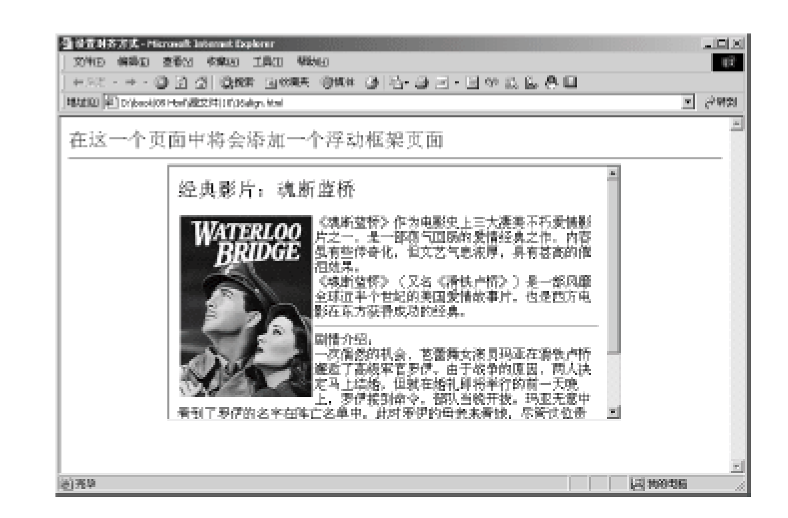


图 10-16　设置对齐方式

10.4.4　共有参数设置

普通框架页面和浮动框架页面所共有的一些参数称为浮动框架的基本属性，主要包括 src、name、marginwidth、marginheight 和 scrolling。由于这些参数的设置方法与普通框架类似，因此下面通过一个实例简单介绍。对于具体的参数可以参照 10.3 节的内容。

语法：

<iframe src="浮动框架页面源文件" name="页面名称" marginwidth="边框与页面的水平距离" marginheight="边框与页面的垂直距离" scrolling="是否显示滚动条">

</iframe>

说明：在这里，这些参数的取值范围与普通框架相同，因此不再重复，只通过下面的实例说明这些属性的设置效果。

实例代码：

```
<html>
<head>
<title>设置框架的共有参数</title>
</head>
<body>
    <font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
    <hr size=2>
    <iframe src="source.html" width="550" height="310" align="center" name="public" marginwidth="60" marginheight="40" scrolling="No">
    </iframe>
</body>
</html>
```

运行这段代码，对比图 10-16，可以看到框架页面的内容与边框增加了一定的空间，而且在页面中虽然无法显示全部的内容，但是并没有出现滚动条，如图 10-17 所示。

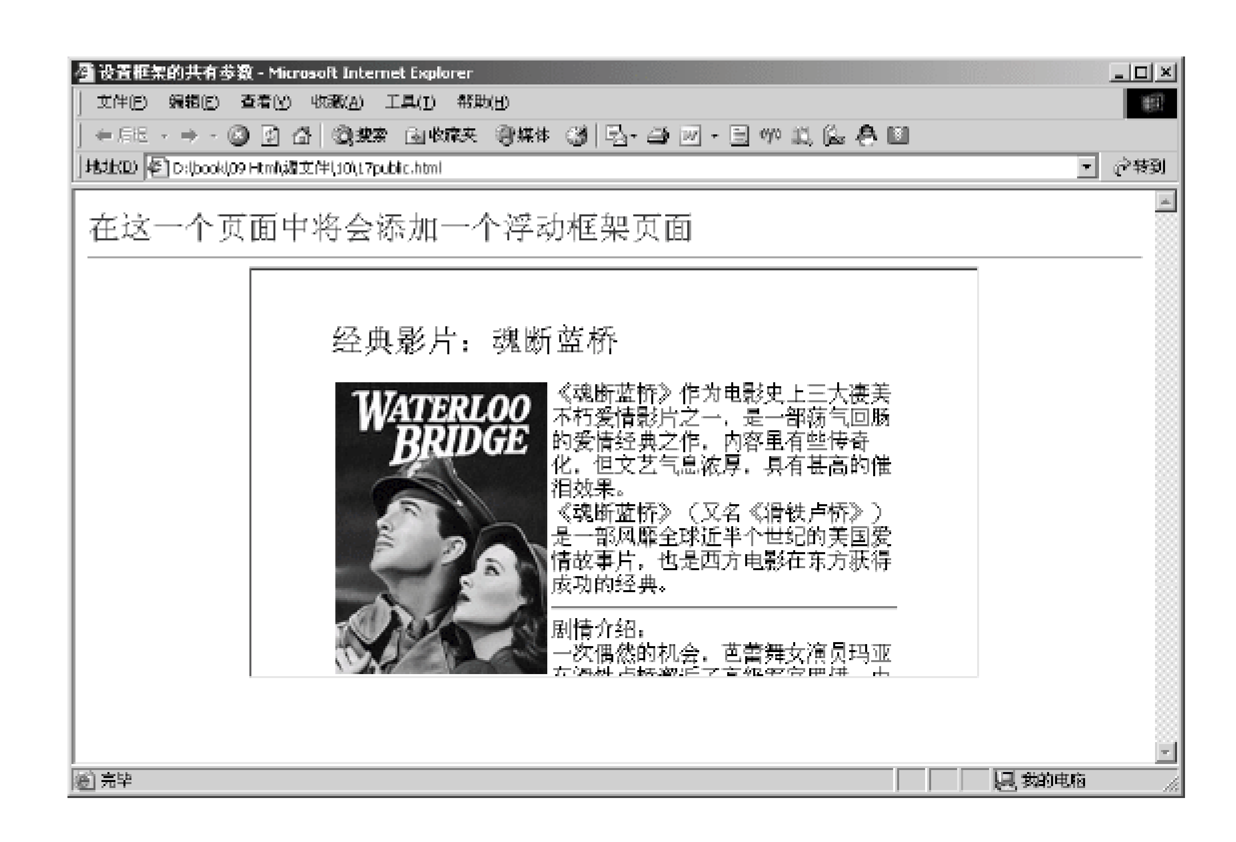


图 10-17　设置框架页面的共有参数

10.4.5　其他参数——frameborder

在浮动框架页面中，还有另外一个特殊的参数可以设置，即框架边框显示属性 frameborder。在普通框架中，该参数既对整个框架集有效，也对单个框架有效。同样地，在浮动框架中也可以设置这一参数。

语法：

```
<iframe src="浮动框架页面源文件" frameborder="是否显示">
</iframe>
```

说明：在这里，这些参数的取值范围与普通框架相同。其中 src 是浮动框架的页面源文件；frameborder 只能取 0 或 1，0 表示不显示边框，1 为默认取值，表示显示边框。下面只通过一个实例说明这些属性的设置效果。

实例代码：

```
<html>
<head>
<title>设置框架页面的边框</title>
</head>
<body>
    <font size="5" color="#CC0000">在这一个页面中将会添加一个浮动框架页面</font>
    <hr size=2>
    <iframe src="source.html" width="550" height="310" align="center" name="public" marginwidth="60 "
marginheight="40" scrolling="No" frameborder="0">
    </iframe>
</body>
</html>
```

运行这段代码，对比图 10-16，可以看到浮动框架的页面边框不见了，如图 10-18 所示。

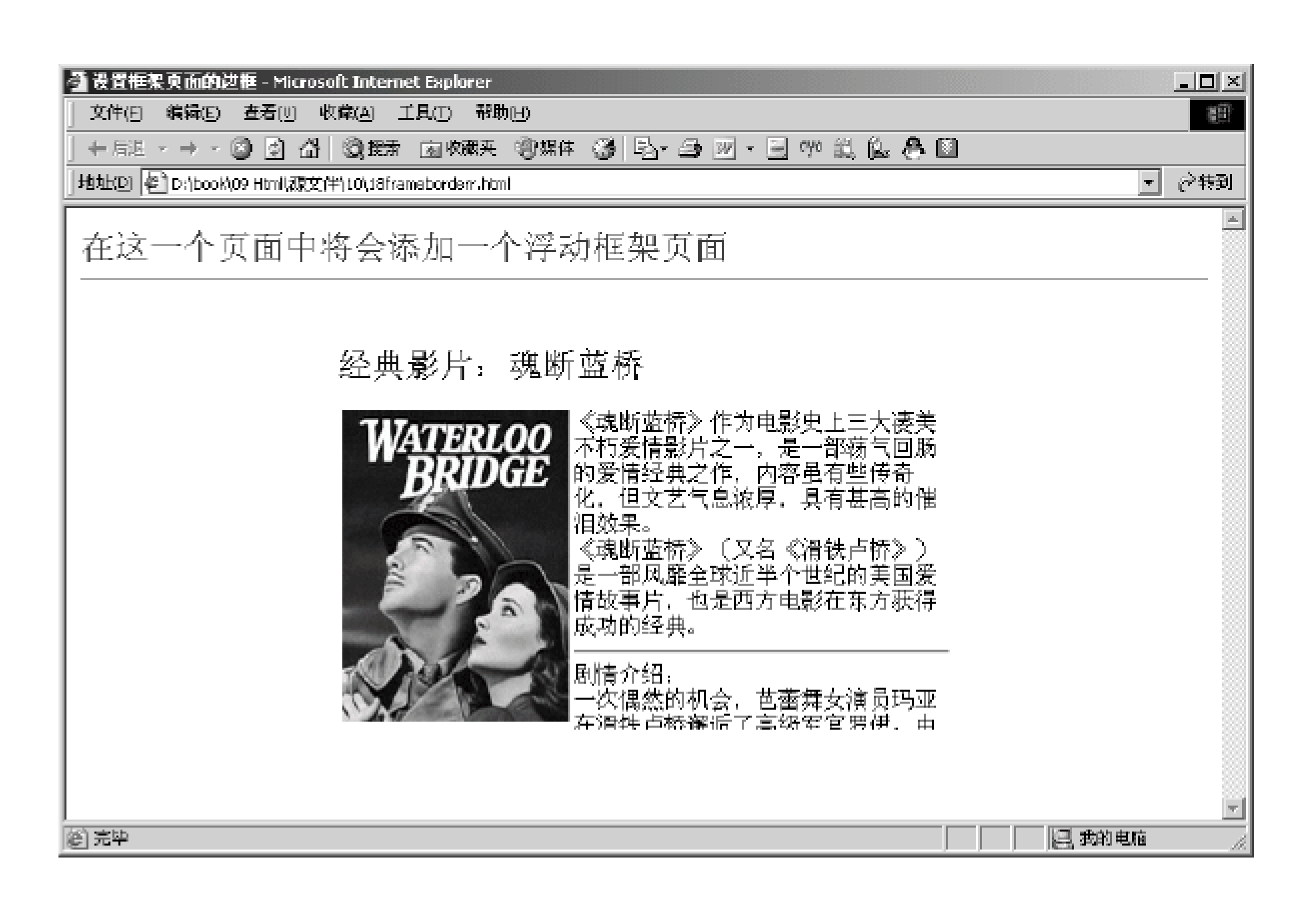


图 10-18　设置框架页面的边框

10.5　框架与链接

大部分情况下，在页面中添加框架是为了更好地对页面内容进行导航，因此常常是通过框架进行链接不同的内容。对于框架页面来说，设置页面的链接需要使用 target 参数，这一参数的取值不同于它在普通 HTML 页面中的取值，下面进行具体的介绍。

10.5.1　设置普通框架结构的链接

普通的框架结构之间通过 target 参数互相链接起来。一般情况下，一个页面中会有一个框架窗口作为导航页面，里面添加了对另外一个框架的内容的链接设置，而这些链接则是通过 target 实现的。

下面通过一个具体的实例来讲解关于框架之间的链接方法。

首先需要设定一个框架集文件，用于确定框架页面的整体布局，这里将文件命名为 19layout.html。19layout.html 的实例代码如下：

```
<html>
<head>
<title>框架页面的整体结构</title>
</head>
<frameset rows="150,*">
    <frame src="navig.html" name="navig" scrolling="No" noresize>
    <frame src="content-1.html" name="content">
</frameset>
</html>
```

此处按照常见网站的上下结构进行布局，将页面上面的框架窗口设置为 150 像素高，下面的窗口则分割剩下的部分。在该文件中，设置了上面的框架窗口页面为导航页面，命名为 navig，同时设置其默认打开的页面名为 navig.html。同时还设置该页面窗口不显示滚动条，而且不允许调整窗口大小。在该文件中还设置了下面的框架窗口为内容页面，命名为 content，设置默认打开的页面名 content-1.html。

下面就介绍这两个文件的具体编码。其中，navig.html 文件将放置在框架的上方，作为页面的导航页，其代码如下：

```
<html>
<head>
<title>导航页面</title>
</head>
<body>
    <center>
      <img src="pic05.jpg"><br><br>
      <a href="content-1.html" target="content">白蛇传</a>
                     
      <a href="content-2.html" target="content">梅妻鹤子</a>
                     
      <a href="content-3.html" target="content">菊花仙子</a>
```

```
    </center>
</body>
</html>
```

在这个文件代码中，target 的值是设置将要显示页面内容的框架名称，此处设置为 content，说明设置的这 3 个内容页面将在 content 框架窗口中显示。

content-1.html 文件是默认的内容页面，其代码如下：

```
<html>
<head>
<title>内容页面</title>
</head>
<body>
    <center>
      <font size=5>白蛇传</font><br><br>
    </center>
          清明时分，西湖岸边花红柳绿，断桥上面游人如梭，真是一幅春光明媚的美丽画面。突然，从西湖底悄悄升上来两个如花似玉的姑娘，怎么回事？人怎么会从水里出来呢？原来，她们是两条修炼成了人形的蛇精，虽然如此，但她们并无害人之心，只因羡慕世间的多彩人生，才一个化名叫白素贞，一个化名叫小青，来到西湖边游玩。<br>
          偏偏老天爷忽然发起脾气来，霎时间下起了倾盆大雨，白素贞和小青被淋得无处藏身，正发愁呢，突然只觉头顶多了一把伞，转身一看，只见一位温文尔雅、白净秀气的年轻书生撑着伞在为她们遮雨。白素贞和这小书生四目相交，都不约而同地红了红脸，相互产生了爱慕之情。小青看在眼里，忙说："多谢！请问客官尊姓大名。"那小书生道："我叫许仙，就住在这断桥边。"白素贞和小青也赶忙作了自我介绍。从此，他们三人常常见面，白素贞和许仙的感情越来越好，过了不久，他们就结为夫妻，并开了一间"保和堂"药店，小日子过得可美了！<br>
          由于"保和堂"治好了很多很多疑难病症，而且给穷人看病配药还分文不收，所以药店的生意越来越红火，远近来找白素贞治病的人越来越多，人们将白素贞亲切地称为白娘子。可是，"保和堂"的兴隆、许仙和白娘子的幸福生活却惹恼了一个人，谁呢？那就是金山寺的法海和尚。因为人们的病都被白娘子治好了，到金山寺烧香求菩萨的人就少多了，香火不旺，法海和尚自然就高兴不起来了。这天，他又来到"保和堂"前，看到白娘子正在给人治病，不禁心内妒火中烧，再定睛一瞧，哎呀！原来这白娘子不是凡人，而是条白蛇变的！<br>
          法海虽有点小法术，但他的心术却不正。看出了白娘子的身份后，他就想拆散许仙白娘子夫妇、搞垮"保和堂"。于是，他偷偷把许仙叫到寺中，对他说："你娘子是蛇精变的，你快点和她分手吧，不然，她会吃掉你的！"许仙一听，非常气愤，他想：我娘子心地善良，对我的情意比海还深。就算她是蛇精，也不会害我，何况她如今已有了身孕，我怎能离弃她呢！法海见许仙不上他的当，恼羞成怒，便把许仙关在了寺里。<br>
           在"保和堂"里，白娘子正焦急地等待许仙回来。一天、两天，左等、右等，白娘子心急如焚。终于打听到原来许仙被金山寺的法海和尚给"留"住了，白娘子赶紧带着小青来到金山寺，苦苦哀求，请法海放回许仙。法海见了白娘子，一阵冷笑，说道："大胆妖蛇，我劝你还是快点离开人间，否则别怪我不客气了！"白娘子见法海拒不放人，无奈，只得拔下头上的金钗，迎风一摇，掀起滔滔大浪，向金山寺直逼过去。法海眼见水漫金山寺，连忙脱下袈裟，变成一道长堤，拦在寺门外。大水涨一尺，长堤就高一尺，大水涨一丈，长堤就高一丈，任凭波浪再大，也漫不过去。再加上白娘子有孕在身，实在斗不过法海，后来，法海使出欺诈的手法，将白娘子收进金钵，压在了雷峰塔下，把许仙和白娘子这对恩爱夫妻活生生地拆散了。<br>
          小青逃离金山寺后，数十载深山练功，最终打败了法海，将他逼进了螃蟹腹中，救出了白娘子，从此，她和许仙以及他们的孩子幸福地生活在一起，再也不分离了。
</body>
</html>
```

用同样的方法添加另外两个内容页面，分别命名为 content-2.html 和 content-3.html，具体的代码见

配套光盘相对应的文件夹，这里不再过多介绍。

完成这几个文件之后，打开框架集页面，效果如图 10-19 所示。

单击页面中的链接文字“梅妻鹤子”，会在框架的下方页面中显示 content-2.html 文件的内容，如图 10-20 所示。

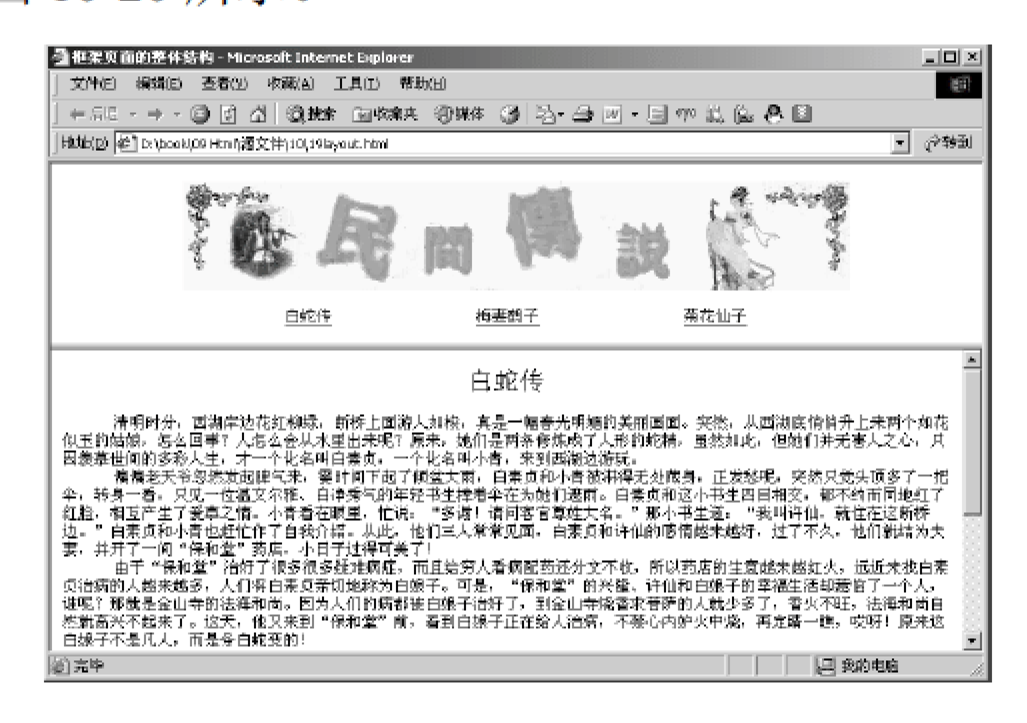


图 10-19　框架页面首页效果

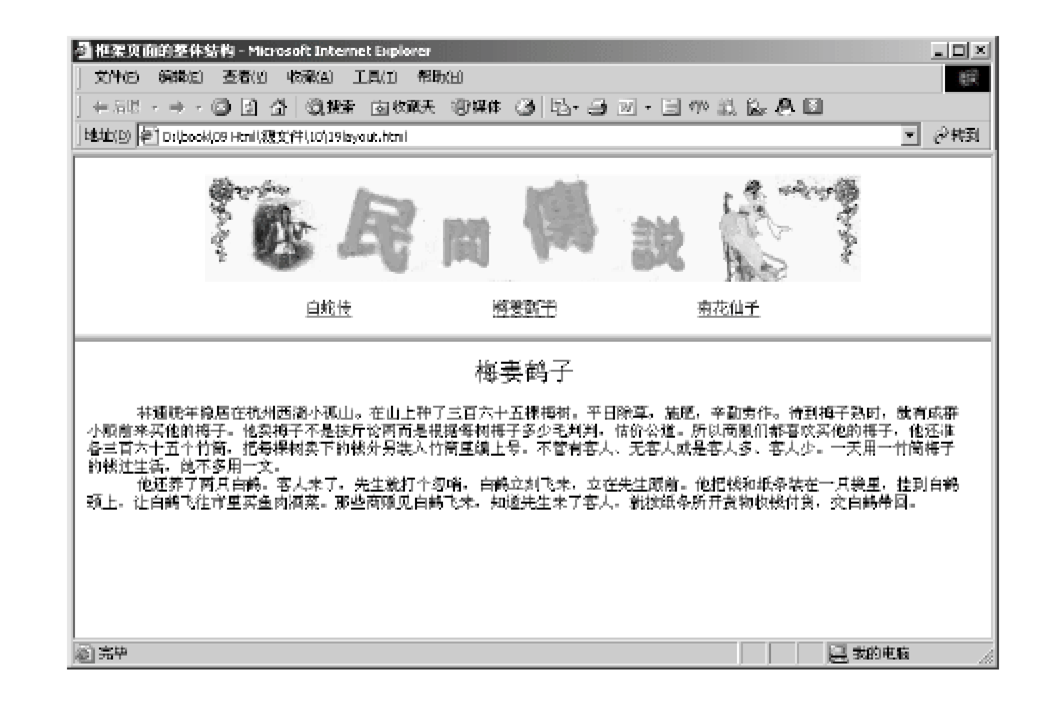


图 10-20　链接到 content-2 页面

单击页面中的链接文字“菊花仙子”，会在框架的下方页面中显示 content-3.html 文件的内容，如图 10-21 所示。这就说明，在框架集中实现了各个窗口页面之间的链接。

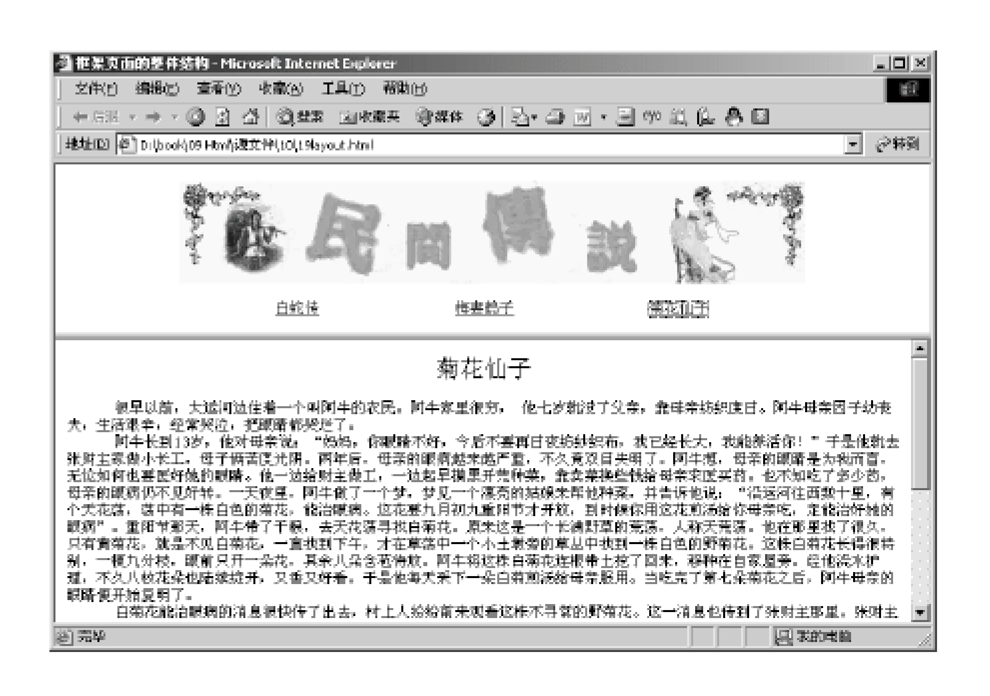


图 10-21　链接到 content-3 页面

10.5.2　浮动框架与链接

在浮动框架中同样也可以制作页面之间的链接，其方法与普通框架类似，下面依然通过实例来说明。

浮动框架所在的主页面命名为 20float.html，代码如下：

```
<html>
<head>
<title>浮动框架窗口的链接</title>
</head>
<body>
    <center>
      <img src="pic05.jpg"><br><br>
      <a href="content-1.html" target="content">白蛇传</a>
```

```
                           
            <a href="content-2.html" target="content">梅妻鹤子</a>
                           
            <a href="content-3.html" target="content">菊花仙子</a>
            <br><br>
        </center>
            <iframe src="content-1.html" name="content" width="500" height="170" align="center">
            </iframe>
    </body>
    </html>
```

运行这段代码，可以看到如图 10-22 所示的效果。

单击其中的链接文字“菊花仙子”，可以在下面的浮动框架页面中打开 content-3.html 文件的内容，如图 10-23 所示。由此可见，实例中成功地实现了浮动框架窗口的链接。

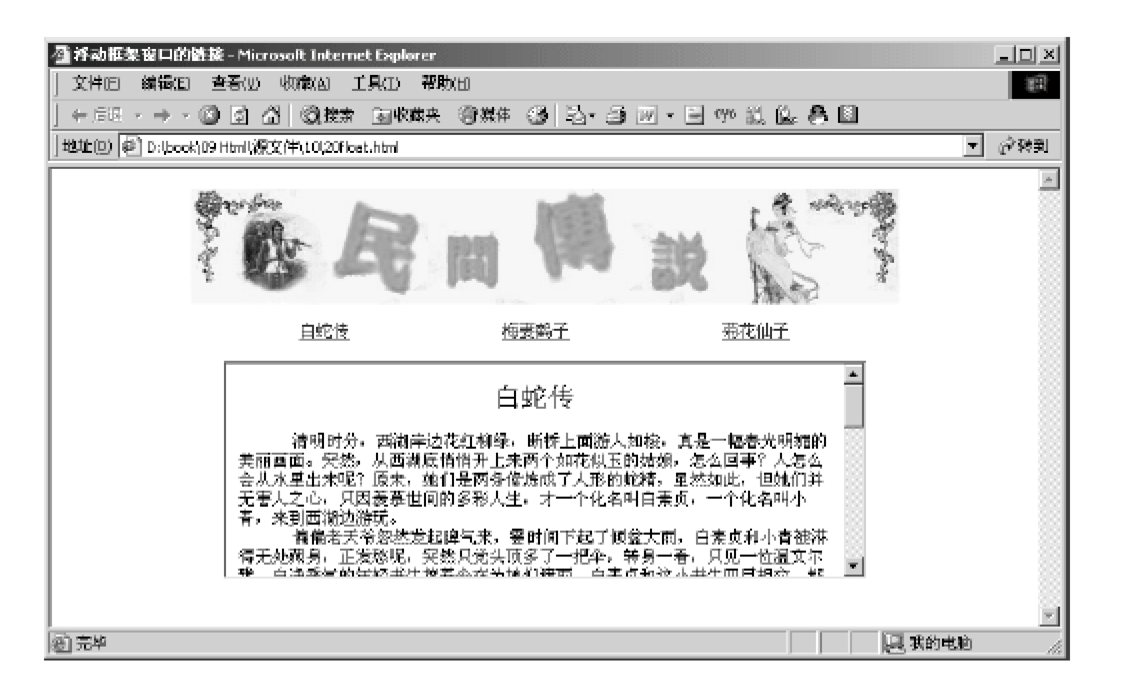


图 10-22　浮动框架窗口的主页面

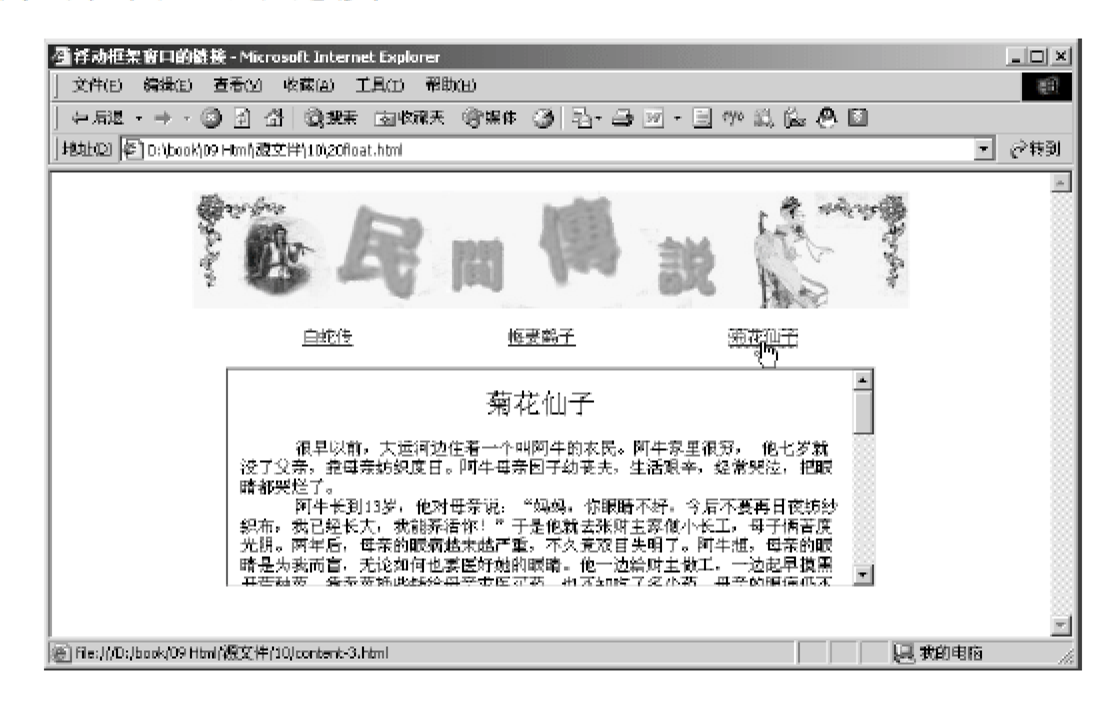


图 10-23　链接 content-3.html 文件的效果

第 11 章

CSS 样式表

- CSS 简介
- CSS 的使用
- 设置 CSS 的字体属性
- 颜色及背景属性
- 文本属性
- 边距与填充属性
- 边框属性
- 定位及尺寸属性
- 列表属性
- 光标属性——cursor
- 滤镜属性

随着网页设计技术的发展，人们已经渐渐不满足于原有的一些 HTML 标记，而是希望能够为页面内容添加一些更加绚丽的属性，例如鼠标标记、滤镜以及渐变效果等。CSS 技术的飞速发展使这些需求成为现实，可以对图片、文本容器和其他一些对象使用 CSS 技术，而且不用编写一行令人费解的代码。本章将详细讲解网页设计中 CSS 的功能和编写方法。

11.1　CSS 简介

11.1.1　CSS 的概念

CSS 是 Cascading Style Sheet 的缩写，可以翻译为“层叠样式表”或“级联样式表”，即“样式表”。CSS 的属性在 HTML 元素中是依次出现的，并不显示在浏览器中。它可以定义在 HTML 文档的标记（tag）里，也可以在外部附加文档中作为外加文件。此时，一个样式表可以作用多个页面，乃至整个站点，因此具有更好的易用性和扩展性。

利用 CSS 不仅可以控制一篇文档中的文本格式，而且可以控制多篇文档的文本格式。因此使用 CSS 样式表定义页面文字，将会使工作量大大减小。一些好的 CSS 样式表的建立，可以更进一步地对页面进行美化，对文本格式进行精确定制。Dreamweaver 还能识别现存文档中已定义的 CSS 样式，方便用户对现有文档进行修改。

CSS 样式表的功能一般可以归纳为以下几点：

- 灵活控制网页中文字的字体、颜色、大小、间距、风格及位置。
- 随意设置一个文本块的行高、缩进，并可以为其加入三维效果的边框。
- 更方便地为网页中的任何元素设置不同的背景颜色和背景图片。
- 精确控制网页中各元素的位置。
- 可为网页中的元素设置各种过滤器，从而产生诸如阴影、辉光、模糊和透明等效果，而这些效果只有在一些图像处理软件中才能实现。
- 可以与脚本语言相结合，使网页中的元素产生各种动态效果。

11.1.2　CSS 的特点

除了可扩展 HTML 的样式设定外，CSS 的特点主要还包含如下几点：

- 减少图形文件的使用：很多网页为求设计效果而大量使用图形，以致网页的下载速度变得很慢。CSS提供了很多的文字样式、滤镜特效等，可以轻松取代原来图形才能表现的视觉效果。这样的设计方式不仅使修改网页内容变得更方便，也大大提高了下载速度。
- 集中管理样式信息：CSS可将网页要展示的内容与样式设定分开，也就是将网页的外观设定信息从网页内容独立出来，并集中管理。这样，当要改变网页外观时，只需更改样式设定的部分，HTML文件本身并不需要更改。
- 共享样式设定：将CSS样式信息存成独立的文件，可以让多个网页文件共同使用，从而避免在每一个网页文件中都要重复设定的麻烦。
- 将样式分类使用：多个HTML文件可使用同一个CSS样式文件，一个HTML网页文件上也可以同时使用多个CSS样式文件。

在同一文本中应用两种或两种以上的样式时，这些样式会相互冲突，产生不可预料的效果。浏览器根据以下规则显示样式属性：

- 如果在同一文本中应用两种样式时，浏览器显示出两种样式中除冲突属性之外的所有属性。
- 如果在同一文本中应用的两种样式是相互冲突的，浏览器显示出最里面的样式属性。
- 如果存在直接冲突，自定义样式的属性（应用CLASS属性的样式）将覆盖HTML标记样式的属性。

11.2　CSS 的使用

11.2.1　CSS 的基本语法

CSS 的基本语法：选择符 { 样式属性: 取值; 样式属性: 取值;…}

其中，选择符可以是多种形式的，例如要定义 HTML 标记中 body 的样式，可以使用如下的代码：

```
body {color: black}
```

这段代码定义了页面主体部分（即 HTML 中<body>标记中的内容）的样式，color 表示主体部分文字的颜色属性，black 是颜色的属性值。因此这段样式代码实现的功能是将页面中的文字显示为黑色。

CSS 样式中的选择符可以有如下几种：

1．类选择符

用类选择符可以把相同的元素分类定义成不同的样式。定义类选择符时，在自定类名称的前面加一个句点（.）。

类选择符的语法：标记名.类名 { 样式属性: 取值; 样式属性: 取值;…}

例如要设置两个文字颜色不同的段落，一个为红色，一个为黑色，可以使用如下代码预定义两个类：

```
p.red { color: red }
p.black { color: black }
```

在这段代码中，定义了段落选择符 p 的 red 和 black 两个类，red 和 black 就可以称为类选择符。类的名称可以是任意英文单词或以英文开头与数字的组合，一般以其功能和效果简要命名。值得注意的是，类选择符在实际应用中，可以省略 HTML 标记名，也就是实例中的 p，直接写成下面的代码：

```
.red { color: red }
.black { color: black }
```

但是与直接定义 HTML 中的标记样式不同的是：这段代码仅仅是定义了样式，并没有应用样式。如果要应用样式中的 red 类，还需要在正文中添加如下代码：

```
<p class=red>
```

同样，如果在页面中要应用 black 类的样式，需要在其中添加如下代码：

```
<p class=black>
```

2．ID 选择符

在 HTML/XML 文档中，在需要惟一标识一个元素时，就会赋予它一个 id 标识，以便在对整个文档进行处理时能够很快地找到这个元素。而 ID 选择符则用来对这个单一元素定义单独的样式。其定义的方法与类选择符大同小异，只需要把句点（.）改成井号（#），而在调用类时需要把 class 换成 id。

ID 选择符的语法：标记名#标识名 { 样式属性: 取值; 样式属性: 取值;…}

例如在页面中定义了一个 id 为 exam 的元素，这里要设置这一元素的文字颜色为红色，那么只需要添加如下代码：

```
#exam { color:#FF0000 }
<p id="exam">
```

这样，页面中包含 id 为 exam 的元素的段落文字显示为红色。但是由于 ID 选择符局限性很大，只能单独定义某个元素的样式，一般只在特殊情况下使用。

3．包含选择符

包含选择符也称为关联选择符，是对某种元素包含关系（例如对元素 1 里包含元素 2）定义的样式表。这种方式只对在元素 1 里的元素 2 定义，对单独的元素 1 或元素 2 无定义，例如：

```
table a { font-size: 12px }
```

这段代码只对在表格内的链接文字有效，设定了文字大小为 12 像素，而对于表格外的链接文字则不起作用。

11.2.2　添加 CSS 的方法

当浏览器读取样式表时，要依照文本格式来读，这里介绍 4 种在页面中插入样式表的方法：链入外部样式表、内部样式表、导入外部样式表和内嵌样式。

1．链入外部样式表

链入外部样式表是把样式表单独保存为一个文件，然后在页面中用<link>标记链接，而这个<link>标记必须放到页面的<head>区域内。

基本语法：

```
<link rel="stylesheet" type="text/css" href="样式表文件的地址">
```

在该语法中，浏览器从样式表文件中以文档格式读出定义的样式表。rel="stylesheet"是指在页面中使用的是外部样式表；type="text/css"是指文件的类型是样式表文本；href 参数则用来设定文件的地址，可以为绝对地址或相对地址。

一个外部样式表文件可以应用于多个页面。当改变这个样式表文件时，所有页面的样式都随之改变。因此常应用在制作大量相同样式页面的网站中，因为它不仅减少了重复的工作量，而且有利于以后的修改、编辑，浏览时也减少了重复下载代码。

样式表文件可以用任何文本编辑器（例如“记事本”）打开并编辑，一般样式表文件扩展名为.css，其内容就是定义的样式，不包含 HTML 标记，例如：

```
hr {color: #0000FF}
p {color:black; font-family: "宋体"}
```

在这段代码中定义了水平线的颜色为蓝色，段落文字的颜色为黑色，字体为宋体。

2．内部样式表

内部样式表是把样式表的内容直接放到页面的<head>区域里，通过<style>标记插入。

基本语法：

```
<style type="text/css">
    选择符 { 样式属性: 取值; 样式属性: 取值;…}
    选择符 { 样式属性: 取值; 样式属性: 取值;…}
    ……
</style>
```

说明：在语法中，关于样式的具体定义给出了这种基本的语法，也可以是前面所说的其他几种，例如类选择器写法等。

由于有些低版本的浏览器不能识别 style 标记，这意味着低版本的浏览器在读取文件时会忽略<style>标记，而把标记中的内容（也就是样式的定义文字）以文本的形式直接显示到页面上。为了避免这样的情况发生，考虑到 HTML 和 CSS 文件的注释方法不同，因此使用 HTML 语言的注释语句（<!-- 注释 -->）隐藏样式表的定义内容。因此实际应用时，内部样式表的语法往往写作：

```
<style type="text/css">
<!--
    选择符 { 样式属性: 取值; 样式属性: 取值;…}
    选择符 { 样式属性: 取值; 样式属性: 取值;…}
    ……
-->
</style>
```

注意： 在本章后面关于设置 CSS 各种属性的讲解中，为了简单明了，均采用内部样式表的方法来讲解。这样，CSS 样式的定义就与 HTML 的内容放置在同一个页面中了。而在实际应用中，为了充分利用 CSS 样式表的优点，还是推荐使用“链入外部样式表”或者下面即将讲解的“导入外部样式表”的方法来进行设置。

3．导入外部样式表

导入外部样式表是指在内部样式表的<style>区域里引用一个外部的样式表文件，导入时需要使用@import 声明。

基本语法：

```
<style type="text/css">
    @import url(样式表的地址)
    选择符 { 样式属性: 取值; 样式属性: 取值;…}
    选择符 { 样式属性: 取值; 样式属性: 取值;…}
    ……
</style>
```

与引用其他文件相同，这里的样式表地址可以是绝对地址，也可以是相对地址。在使用中，需要注意的是导入外部样式表，也就是@import 声明必须在样式表定义的开始部分，而其他样式表的定义都要在@import 声明之后。

4．内嵌样式

内嵌样式是混合在 HTML 标记里使用的，用这种方法可以很直观地对某个元素直接定义样式。内嵌样式的使用是直接在 HTML 标记里加入<style>参数，在<style>参数的内容就是 CSS 的属性和属性值。

基本语法：

```
<HTML 标记 style="样式属性: 取值; 样式属性: 取值;…">
```

在这里，HTML 标记就是页面中标识 HTML 元素的标记，例如 body、p 等。由语法可以很明显地看出：style 参数后面的引号里的内容就相当于样式表大括号里的内容。需要指出的是，style 参数可以应用于 HTML 文件中 body 标记内除了 basefont、param 和 script 之外的任意元素（包括 body 本身）。

11.2.3　CSS 的继承

CSS 的继承性，也被称为样式表的层叠性。样式表的继承规则是外部的元素样式会保留下来继承给这个元素所包含的其他元素。事实上，所有在元素中嵌套的元素都会继承外层元素指定的属性值，有时会把很多层嵌套的样式叠加在一起，除非另外更改。例如，在页面的开始定义了样式表的内容如下：

```
body { color: red; font-size:9pt}
```

而在页面的主体部分有如下的代码：

```
<body>
    ……
    <p>段落文字</p>
    ……
</body>
```

在这个实例中，p 元素包含在元素 body 中，因此标记<p>中的内容会继承 body 定义的属性，也就是实例中的段落文字同样会以红色的 9 像素大小的文字显示。

但是在有些特殊的情况下，内部选择符不继承包含它的选择符的属性值。例如，上边界属性值是不会继承的，因为一般情况认为段落不会同文档的 body 有同样的上边界值。

11.2.4　CSS 样式的冲突

如果在同一个选择器上使用几个不同的样式表时，这个属性值将会叠加几个样式表，遇到冲突的地方会以最后定义的为准。例如在 11.2.3 节的实例中，如果同时定义了 p 的颜色，也就是样式表的内容更改为：

```
body { color: red; font-size:9pt}
p { color:blue}
```

那么在页面显示时，段落文字的字号会继承 body 的 9 像素，而颜色则按照最后定义的蓝色显示。

不同的选择符定义相同的元素时，要考虑到不同的选择符之间的优先级。ID 选择符、类选择符和 HTML 标记选择符，因为 ID 选择符是最后加到元素上的，所以优先级最高，其次是类选择符。例如：

```
p { color: 黑色}
```

```
.blue { color: 蓝色}
#id1 { color: 红色}
```

如果对页面中的一个段落同时设置这 3 种样式，那么会按照优先权最高的 ID 选择符显示为红色的文字。

依照后定义的优先这一原则，一般认为优先级最高的是内嵌样式，内部样式表高于导入外部样式表，链入外部样式表和内部样式表之间则根据定义的先后顺序来评定，也就是最后定义的优先级高。

11.2.5　书写 CSS 的注意事项

一般情况下，在书写 CSS 样式表时，需要注意以下原则：

（1）如果属性的值是多个单词组成，则必须使用引号（""）将属性值括起来。

（2）如果要对一个选择符指定多个属性，则需要使用分号（;）将所有的属性和值分开。

（3）可以将具有相同属性和属性值的选择符组合起来，用逗号（,）将其分开，这样可以减少样式的重复定义。例如，要定义段落和表格内的文字尺寸为 9 像素，则可以使用下面这段代码：

```
p, table{ font-size: 9pt }
```

而这段代码的效果完全等效于对这两个选择符分别定义：

```
p { font-size: 9pt }
table { font-size: 9pt }
```

（4）CSS 样式表中的注释语句以：“/*” 开头，以“*/” 结尾，如下所示：

```
/* 定义段落的样式表 */
p
 {
color: black; font-family: arial /* 颜色为黑色，字体为 arial */
}
```

11.3　设置 CSS 的字体属性

从本节开始，将系统地介绍 CSS 的各种属性，因为对页面产生影响的最终手段还是使用属性。因此，熟练掌握 CSS 的每一个属性及各个属性值是十分必要的。

CSS 的字体属性主要包括字体族科、字体大小、字体风格、加粗字体以及英文字体的大小写转换。

11.3.1　设置字体——font-family

字体族科实际上就是在 CSS 中设置文字的字体，用于改变 HTML 标志或元素的字体，相当于 HTML 标记中 font-face 属性的功能。

语法：font-family: "字体 1","字体 2",…

说明：如果在 font-family 属性中定义了不只一个字体，那么浏览器会由前向后选用字体。也就是

说，当浏览器不支持第一个字体时，会采用第二个字体；如果前两个字体都不支持，则采用第三个字体，以此类推。如果浏览器不支持定义的所有字体，则会采用系统的默认字体。同时需要注意的是，必须用双引号引住任何包含空格的字体名。

实例代码：

```
<html>
<head>
<title>设置字体</title>
<style>
<!--
	h2 { font-family: "华文彩云","黑体"}
	.text { font-family: "宋体"}
-->
</style>
</head>
<body>
	<h2>庐山美景——小天池</h2>
	<p class=text>    小天池位于庐山牯岭北面，池中之水置于高山而终年不溢不涸。池后山脊上，屹立着一座白塔似的喇嘛塔。塔建于 1936 年。 小天池山对面还有一怪石，远望似一雄鹰伸颈欲鸣。鹰首有巨石叠就，一石伸出鹰嘴崖，石缝中绿树芳草婆娑似羽毛，名鹞鹰嘴。山南麓有一峰凌空突出，下临深壑。峰顶悬崖上有一伞顶圆亭。步入亭中，长江似一条白色缎带，飘忽在天际。山脚是深谷，形如一把打开的剪刀，名剪刀峡。峡中溪流淙淙，怪石嶙峋，幽雅秀美。 小天池是牯岭观日出、晚霞、云海等奇观最佳的地方之一。登临山顶，好似坐上一架直升飞机，盘桓在庐山的上空，青山碧水，红瓦苍崖，历历在目，令人纵横浮想，思绪万千。</p>
</body>
</html>
```

运行代码，可以看到页面中应用了在<style>标记中设定的字体，其中<h2>标记中的文字以定义的华文彩云显示，段落中的文字则以 text 类中定义的宋体显示，如图 11-1 所示。当浏览器不支持华文彩云字体时，<h2>标记中的文字则会以黑体显示。

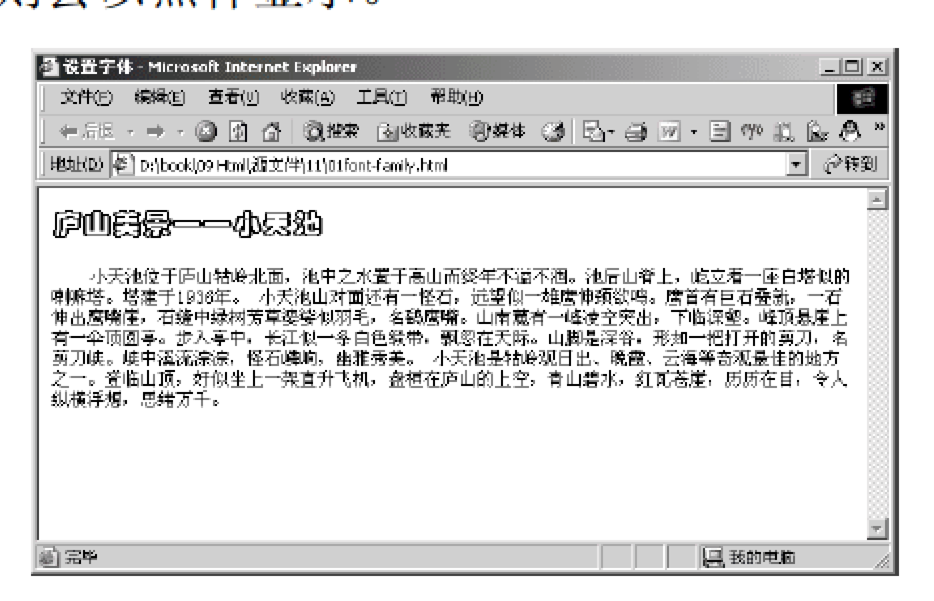


图 11-1　给页面文字指定字体

11.3.2　设置字号——font-size

字体大小属性用作修改字体显示的大小。

语法：font-size:大小取值

取值范围：

- 绝对大小：xx-small | x-small | small | medium | large | x-large | xx-large。
- 相对大小：larger | smaller。
- 长度值或百分比。

说明：绝对大小根据对象字体进行调节，包括 xx-small | x-small | small | medium | large | x-large | xx-large。相对大小则是相对于父对象中字体尺寸进行相对调节，它使用成比例的 em 单位计算，可设置为 larger 或 smaller。长度则是由浮点数字和单位标识符组成的长度值，使用的单位可以为 pt（点，1 点=1/72 英寸）、px（像素）、in（英寸）等，其中浮点数字不能为负值。百分比取值基于父对象中字体的尺寸。

实例代码：

```
<html>
<head>
<title>设置字体</title>
<style>
<!--
    h2 { font-family: "华文彩云","黑体"; font-size:12pt}
    .text { font-family: "宋体"; font-size:12px}
-->
</style>
</head>
<body>
    <h2>庐山美景——小天池</h2>
    <p class=text>    小天池位于庐山牯岭北面，池中之水置于高山而终年不溢不涸。
池后山脊上，屹立着一座白塔似的喇嘛塔。塔建于 1936 年。 小天池山对面还有一怪石，远望似一雄鹰伸颈欲鸣。
鹰首有巨石叠就，一石伸出鹰嘴崖，石缝中绿树芳草婆娑似羽毛，名鹞鹰嘴。山南麓有一峰凌空突出，下临深壑。
峰顶悬崖上有一伞顶圆亭。步入亭中，长江似一条白色缎带，飘忽在天际。山脚是深谷，形如一把打开的剪刀，
名剪刀峡。峡中溪流淙淙，怪石嶙峋，幽雅秀美。 小天池是牯岭观日出、晚霞、云海等奇观最佳的地方之一。登
临山顶，好似坐上一架直升飞机，盘桓在庐山的上空，青山碧水，红瓦苍崖，历历在目，令人纵横浮想，思绪万
千。</p>
</body>
</html>
```

运行这段代码，可以看到<h2>标记中的文字字号被设定为 12pt，即 12 点；而段落中的文字则显示为 12px，即 12 像素，如图 11-2 所示。从图中很明显可以看出 12pt 和 12px 的大小相差很多。

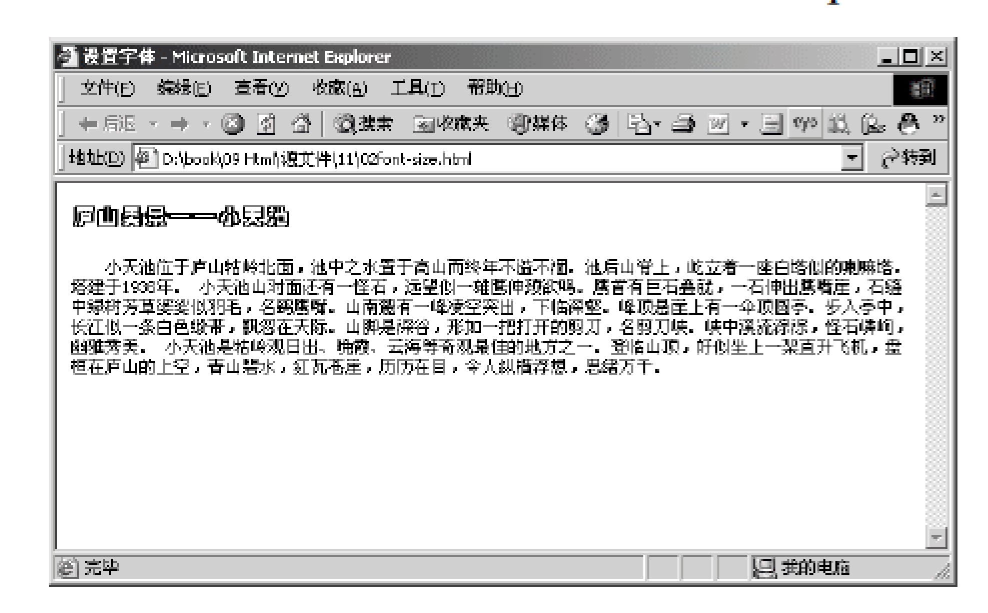


图 11-2　设置字号

在设置字号时，还可以同时设置文字的行高，其取值方法和单位与设置字号相同，如设置带行高

的字号可以是 font-size:12pt/14pt，这就表示设置的文字是 12 点大小，行高是 14 点。

11.3.3　字体风格——font-style

字体风格就是字体样式，主要是设置字体是否为斜体。

语法：font-style:样式的取值

样式的取值范围：normal | italic | oblique

说明：normal 是以正常的方式显示；italic 则是以斜体显示文字；oblique 属于其中间状态，以偏斜体显示。

实例代码：

```
<html>
<head>
<title>设置字体</title>
<style>
<!--
    h2 { font-family: "华文彩云","黑体"; font-size:12pt; font-style:italic}
    .text { font-family: "宋体"; font-size:12px; font-style:oblique}
-->
</style>
</head>
<body>
    <h2>庐山美景——小天池</h2>
    <p class=text>    小天池位于庐山牯岭北面，池中之水置于高山而终年不溢不涸。
池后山脊上，屹立着一座白塔似的喇嘛塔。塔建于 1936 年。 小天池山对面还有一怪石，远望似一雄鹰伸颈欲鸣。
鹰首有巨石叠就，一石伸出鹰嘴崖，石缝中绿树芳草婆娑似羽毛，名鹞鹰嘴。山南麓有一峰凌空突出，下临深壑。
峰顶悬崖上有一伞顶圆亭。步入亭中，长江似一条白色缎带，飘忽在天际。山脚是深谷，形如一把打开的剪刀，
名剪刀峡。峡中溪流淙淙，怪石嶙峋，幽雅秀美。 小天池是牯岭观日出、晚霞、云海等奇观最佳的地方之一。登
临山顶，好似坐上一架直升飞机，盘桓在庐山的上空，青山碧水，红瓦苍崖，历历在目，令人纵横浮想，思绪万
千。</p>
</body>
</html>
```

运行这段代码，可以看到<h2>标记中的文字以斜体的方式显示；而段落中的文字则以偏斜体显示，效果如图 11-3 所示。

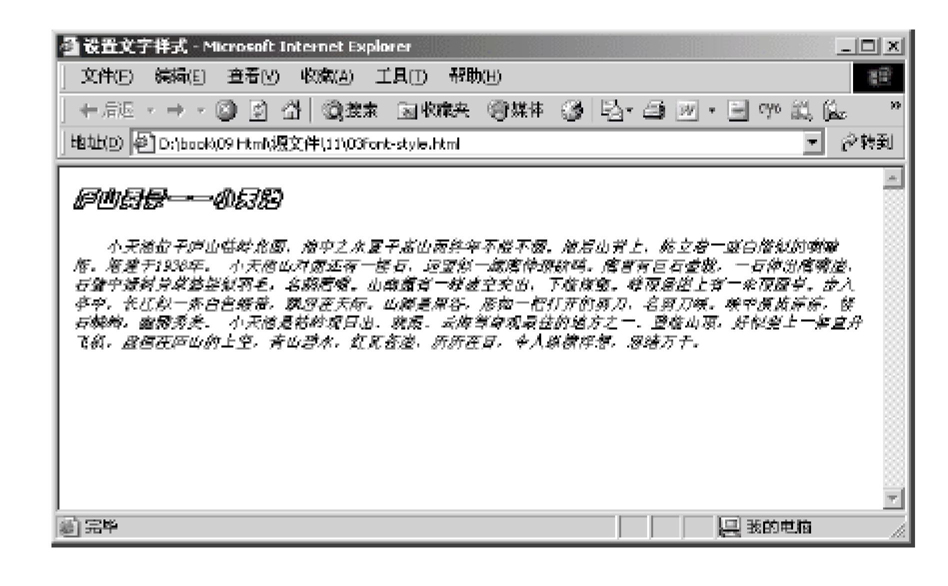


图 11-3　设置文字的样式

11.3.4　设置加粗字体——font-weight

font-weight 属性用于设置字体的粗细，实现对一些字体的加粗显示。

语法：font-weight: 字体粗度值

取值范围：normal | bold | bolder | lighter | number

说明：font-weight 的取值可以是其中的任何一种。其中 normal 表示正常粗细；bold 表示粗体，bolder 则表示特粗体，就是在粗体的基础上再加粗；lighter 表示特细体，就是比正常字体还要细；number 不是真正的取值，它表示 font-weight 还可以取数值，其范围是 100~900，而且一般情况下都是整百的数字，如 100、200 等。正常字体相当于取数值 400 的粗细；粗体则相当于 700 的粗细。

实例代码：

```
<!--设置字体的粗细-->
<html>
<head>
<title>设置字体</title>
<style>
<!--
      h2 { font-family: "黑体"; font-size:12pt; font-weight:bolder}
      .text { font-family: "宋体"; font-size:12px; font-weight:200}
-->
</style>
</head>
<body>
      <h2>庐山美景——小天池</h2>
      <p class=text>    小天池位于庐山牯岭北面，池中之水置于高山而终年不溢不涸。池后山脊上，屹立着一座白塔似的喇嘛塔。塔建于 1936 年。 小天池山对面还有一怪石，远望似一雄鹰伸颈欲鸣。鹰首有巨石叠就，一石伸出鹰嘴崖，石缝中绿树芳草婆娑似羽毛，名鹞鹰嘴。山南麓有一峰凌空突出，下临深壑。峰顶悬崖上有一伞顶圆亭。步入亭中，长江似一条白色缎带，飘忽在天际。山脚是深谷，形如一把打开的剪刀，名剪刀峡。峡中溪流淙淙，怪石嶙峋，幽雅秀美。 小天池是牯岭观日出、晚霞、云海等奇观最佳的地方之一。登临山顶，好似坐上一架直升飞机，盘桓在庐山的上空，青山碧水，红瓦苍崖，历历在目，令人纵横浮想，思绪万千。</p>
</body>
</html>
```

运行程序，可以看到效果如图 11-4 所示。其中<h2>标记中的文字采用了特粗字体，相当于 900 的粗细；在段落文字中采用了 200 的粗细，比正常字体还要细一些。

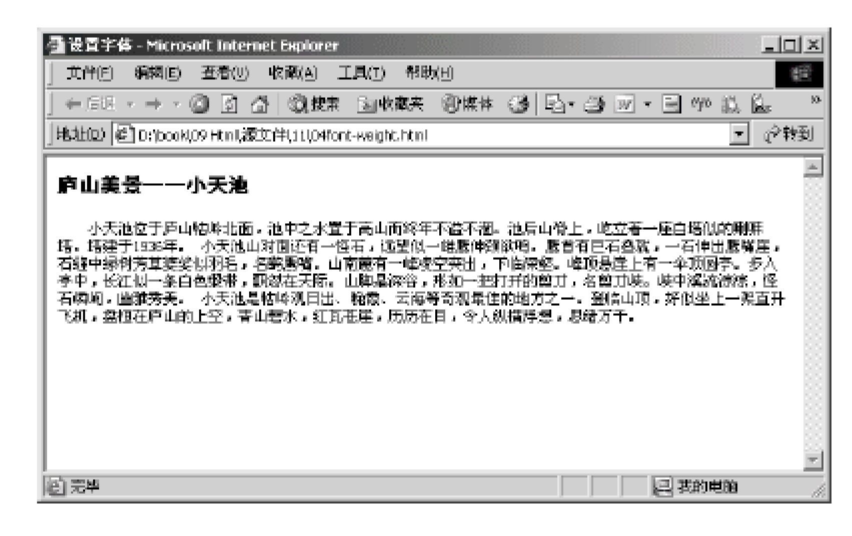


图 11-4　设置字体的粗细

注意：因为不是所有的字体都有 100 ~ 900 这样 9 个有效的加粗显示，因此一些字体的粗细在某些时候会被替代。例如，当设置了 font-weight 的取值为 500，而该字体没有 500 这一个档次的粗细，那么它就会以 400 的粗细显示。

11.3.5　小型的大写字母——font-variant

font-variant 属性用来设置英文字体是否显示为小型的大写字母。

语法：font-variant :取值

取值范围：normal | small-caps

说明：normal 表示正常的字体，small-caps 表示英文显示为小型的大写字母字体。

实例代码：

```
<html>
<head>
<title>设置小型的大写字母</title>
<style>
<!--
      h2 { font-family: "黑体"; font-size:18pt; font-weight:bolder; font-variant: small-caps }
      .text { font-family: "Times New Roman"; font-size:14pt; font-variant: normal}
      .newtext { font-family: "Times New Roman"; font-size:14pt; font-variant: small-caps}
-->
</style>
</head>
<body>
      <h2> contrast </h2>
      <p class=text>The craftsman, Arnold, came from a family of carpenters. </p>
      <p class=newtext>The craftsman, Arnold, came from a family of carpenters. </p>
</body>
</html>
```

运行代码，可以看到同样一句话，通过 font-variant 设置了不同的效果，如图 11-5 所示。

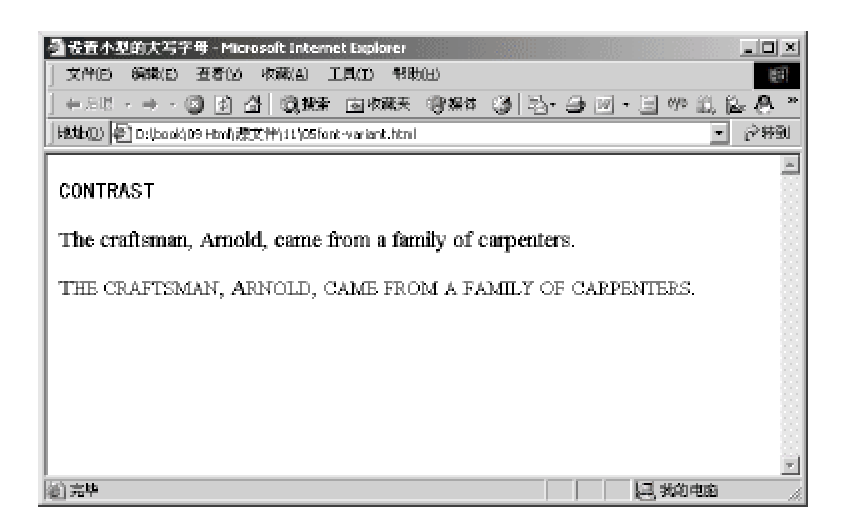


图 11-5　设置小型大写字母

11.3.6　复合属性：字体——font

font 属性是复合属性，用作对不同字体属性的略写，特别是行高。

语法：font:字体取值

说明：在这里，字体取值可以包含字体族科、字体大小、字体风格、加粗字体、小型大写字母，之间用空格相连。例如可以设置为 p { font: italic small-caps bold 12pt/18pt 宋体; }，表示设置的段落文字是粗体和斜体的宋体，大小是 12 点、行高是 18 点，文字中的英文采用小型的大写字母。

实例代码：

```
<!--使用 font 属性设置字体-->
<html>
<head>
<title>设置字体</title>
<style>
<!--
	h2 {font:bolder italic 14pt 黑体}
	.text {font:200 10pt/14pt 宋体}
-->
</style>
</head>
<body>
	<h2>庐山美景——小天池</h2>
	<p class=text>    小天池位于庐山牯岭北面，池中之水置于高山而终年不溢不涸。池后山脊上，屹立着一座白塔似的喇嘛塔。塔建于 1936 年。 小天池山对面还有一怪石，远望似一雄鹰伸颈欲鸣。鹰首有巨石叠就，一石伸出鹰嘴崖，石缝中绿树芳草婆娑似羽毛，名鹞鹰嘴。山南麓有一峰凌空突出，下临深壑。峰顶悬崖上有一伞顶圆亭。步入亭中，长江似一条白色缎带，飘忽在天际。山脚是深谷，形如一把打开的剪刀，名剪刀峡。峡中溪流淙淙，怪石嶙峋，幽雅秀美。 小天池是牯岭观日出、晚霞、云海等奇观最佳的地方之一。登临山顶，好似坐上一架直升飞机，盘桓在庐山的上空，青山碧水，红瓦苍崖，历历在目，令人纵横浮想，思绪万千。</p>
</body>
</html>
```

运行这段代码，效果如图 11-6 所示。

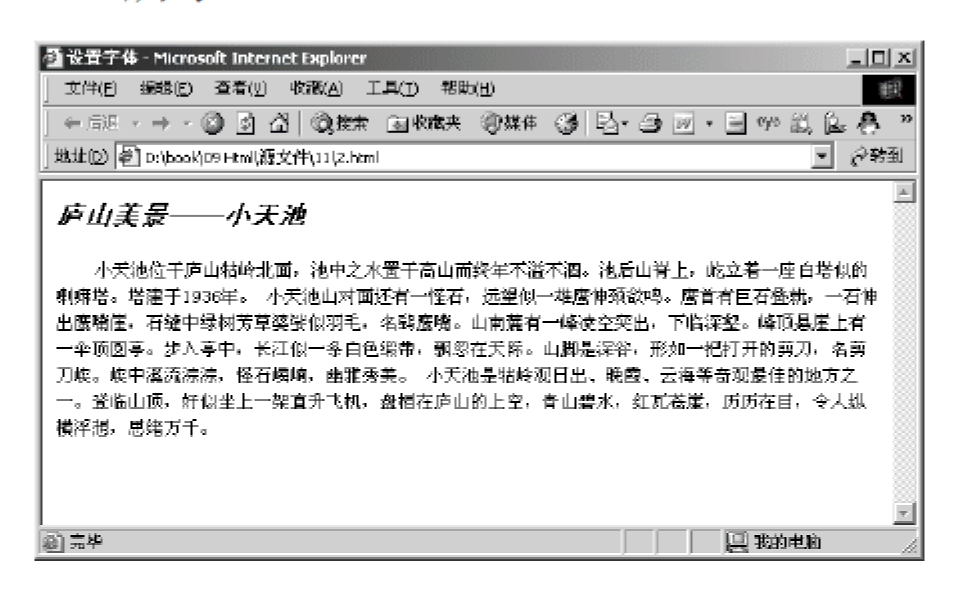


图 11-6　使用 font 属性简略设置字体

11.4　颜色及背景属性

颜色和背景属性主要包括颜色属性设置、背景颜色、背景图像、背景重复、背景附件和背景位置。

11.4.1　颜色属性设置——color

颜色属性允许网页制作者指定一个元素的颜色。颜色值是一个关键字或一个 RGB 格式的数字。常用的预设颜色关键字有 16 个，具体见表 11-1。

表11-1　常用的16种颜色关键字

颜色关键字	中 文 含 义	十六进制RGB值
aqua	水绿色	#00FFFF
black	黑色	#000000
blue	蓝色	#0000FF
fuchsia	紫红色	#FF00FF
gray	灰色	#808080
green	绿色	#008000
lime	酸橙色	#00FF00
maroon	栗色	#800000
navy	海军蓝	#000080
olive	橄榄色	#808000
purple	紫色	#800080
red	红色	#FF0000
silver	银色	#C0C0C0
teal	水鸭色	#008080
white	白色	#FFFFFF
yellow	黄色	#FFFF00

为了避免与用户的样式表之间的冲突，建议颜色和背景属性始终一起指定。

语法：color:颜色取值

说明：在这里颜色取值可以是颜色关键字，如 yellow，也可以是 RGB 颜色代码。在 CSS 中，RGB 颜色代码有多种写法：

- #rrggbb：如#FF0000。
- #rgb：如#F00。
- rgb(x,x,x)：其中，x是一个介于0~255之间的整数，例如rgb(255,0,0)。
- rgb(y%,y%,y%)：其中，y是一个介于0.0~100.0之间的整数，例如，rgb(100%,0%,0%)。

这些颜色代码都表示红色，即关键字为red的颜色。

实例代码：

```
<html>
<head>
<title>设置颜色</title>
<style>
<!--
	h2 { font-family:"黑体"; font-size:15pt; color:red}
	.text { font-family:"宋体"; font-size:10pt; color:#8B0000 }
```

```
-->
</style>
</head>
<body>
    <h2>梅花——花中君子</h2>
    <p class=text>     梅花树的枝干不是很粗，但长得很独特。它们很少像其他树的枝干那样笔直地伸长开，而多是曲曲折折、盘旋而上。每年大地将要复苏时，深褐色的老枝上便会抽出一些挺拔的新枝。这些嫩绿细小的新枝争先恐后地往外伸去，像是比赛似的。到了初春时分，这些新枝上就会爬满一个个含苞待放的花骨朵。只等一夜春风吹来，这些个小花骨朵便会在风中绽开美丽的笑脸，用各自迷人的身姿装扮春天的大地。</p>
    <p class=text>     梅花的种类很多，有白如雪的白梅；有粉如霞的宫粉梅；有晶莹如玉的绿萼梅；有火红似烈焰的朱砂梅。从花形上来看，梅花分为单瓣和复瓣，单瓣的梅花除了细如发丝的花蕊之外，周围只有五片花瓣，可是复瓣梅有时一朵花会有十几片花瓣……</p>
</body>
</html>
```

运行这段代码，可以看到两种文字设置了不同的颜色，如图 11-7 所示。其中，<h2>标记中的文字显示为红色；而段落文字则以#8B0000 颜色显示。

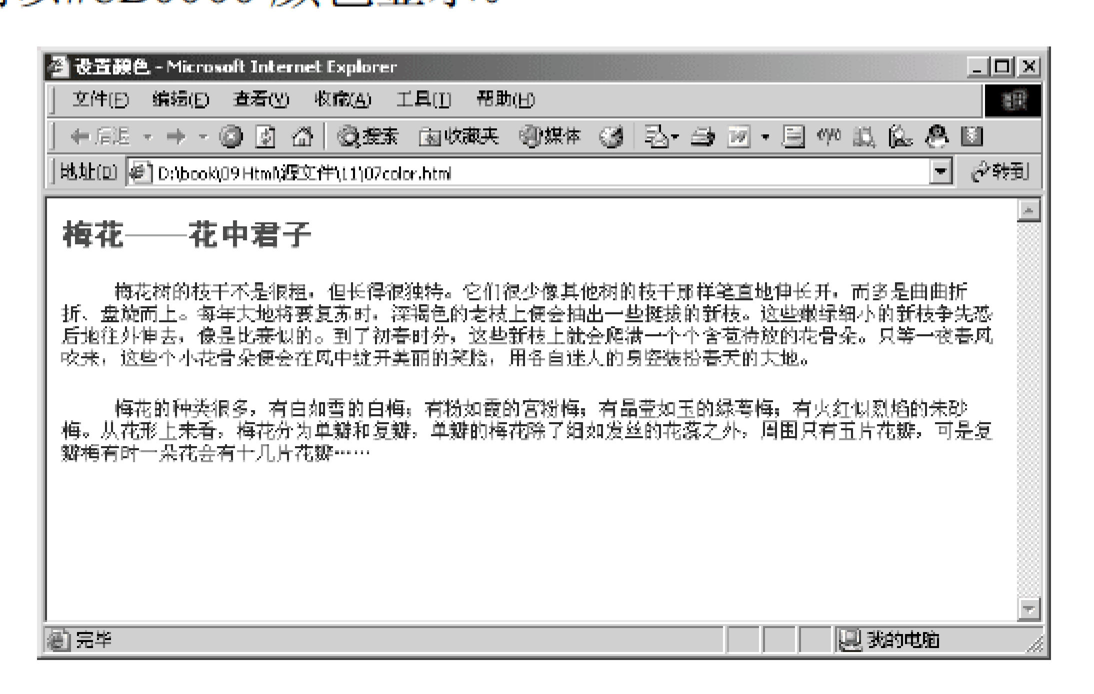


图 11-7　设置颜色

11.4.2　背景颜色——background-color

在 CSS 中，使用 background-color 属性设置背景颜色。

语法：background-color:颜色取值

实例代码：

```
<html>
<head>
<title>设置背景颜色</title>
<style>
<!--
    h2 { font-family:"黑体"; font-size:15pt; color:yellow; background-color:fuchsia }
    body {background-color: #CEDDF4}
    .text { font-family:"宋体"; font-size:10pt; color:#8B0000 }
-->
</style>
</head>
```

```
<body>
    <h2>梅花——花中君子</h2>
    <p class=text>     梅花树的枝干不是很粗，但长得很独特。它们很少像其他树的枝干那样笔直地伸长开，而多是曲曲折折、盘旋而上。每年大地将要复苏时，深褐色的老枝上便会抽出一些挺拔的新枝。这些嫩绿细小的新枝争先恐后地往外伸去，像是比赛似的。到了初春时分，这些新枝上就会爬满一个个含苞待放的花骨朵。只等一夜春风吹来，这些个小花骨朵便会在风中绽开美丽的笑脸，用各自迷人的身姿装扮春天的大地。</p>
    <p class=text>     梅花的种类很多，有白如雪的白梅；有粉如霞的宫粉梅；有晶莹如玉的绿萼梅；有火红似烈焰的朱砂梅。从花形上来看，梅花分为单瓣和复瓣，单瓣的梅花除了细如发丝的花蕊之外，周围只有五片花瓣，可是复瓣梅有时一朵花会有十几片花瓣……</p>
</body>
</html>
```

运行这段代码，看到页面中标题文字和页面的<body>主体部分分别设置了不同的背景色，如图 11-8 所示。

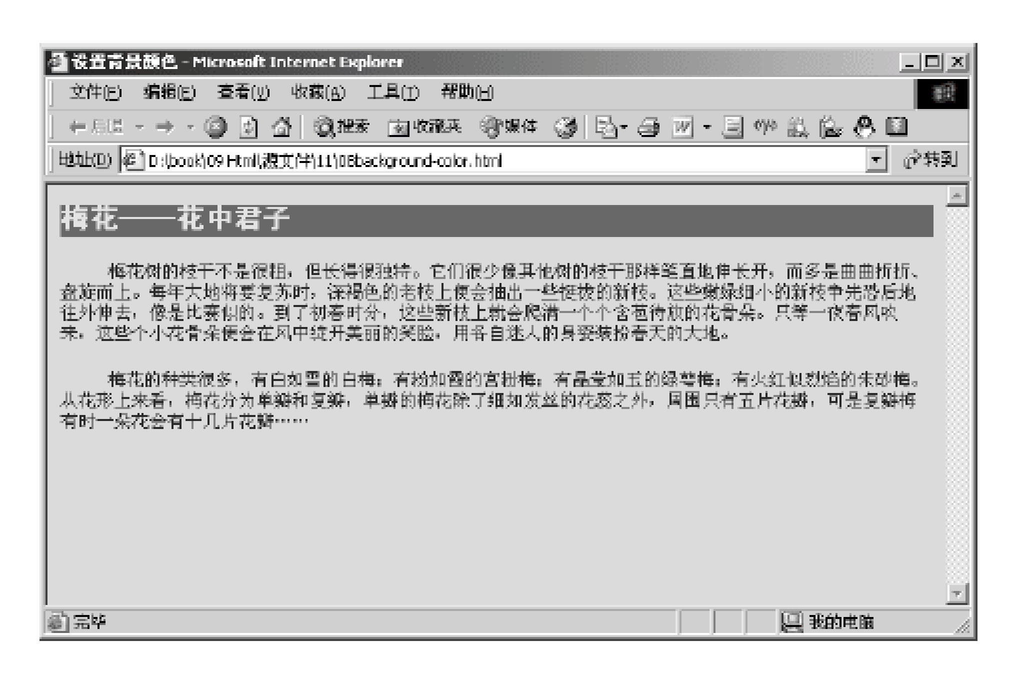


图 11-8　设置文字的背景色

11.4.3　背景图像——background-image

在 CSS 中，背景图像属性为 background-image，用来设定一个元素的背景图像。

语法：background-image: url(图像地址)

说明：在该语法中，图像地址是作为背景的图像文件的地址，可以设置成绝对地址，也可以设置成相对地址。背景图片一般是 JPG、PNG 或者 GIF 格式。

实例代码：

```
<html>
<head>
<title>设置背景图像</title>
<style>
<!--
    h2 { font-family:"黑体"; font-size:15pt; color:yellow; background-color:fuchsia }
    body {background-color:#CEDDF4; background-image:url(pic01.jpg)}
    .text { font-family:"宋体"; font-size:10pt; color:#8B0000 }
-->
</style>
</head>
```

```
<body>
    <h2>梅花——花中君子</h2>
    <p class=text>     梅花树的枝干不是很粗，但长得很独特。它们很少像其他树的枝干那样笔直地伸长开，而多是曲曲折折、盘旋而上。每年大地将要复苏时，深褐色的老枝上便会抽出一些挺拔的新枝。这些嫩绿细小的新枝争先恐后地往外伸去，像是比赛似的。到了初春时分，这些新枝上就会爬满一个个含苞待放的花骨朵。只等一夜春风吹来，这些个小花骨朵便会在风中绽开美丽的笑脸，用各自迷人的身姿装扮春天的大地。</p>
    <p class=text>     梅花的种类很多，有白如雪的白梅；有粉如霞的宫粉梅；有晶莹如玉的绿萼梅；有火红似烈焰的朱砂梅。从花形上来看，梅花分为单瓣和复瓣，单瓣的梅花除了细如发丝的花蕊之外，周围只有五片花瓣，可是复瓣梅有时一朵花会有十几片花瓣……</p>
</body>
</html>
```

运行这段代码，可以看到页面效果如图 11-9 所示。

为了避免那些设定了不载入图像的浏览器无法显示相应的效果这一问题，一般情况下当定义了背景图像后，应该也定义一个类似的背景颜色。而当同时设定了背景颜色和背景图像时，会优先显示背景图像属性。在本例中，如果将浏览器设定为不载入图像，则显示效果如图 11-10 所示。

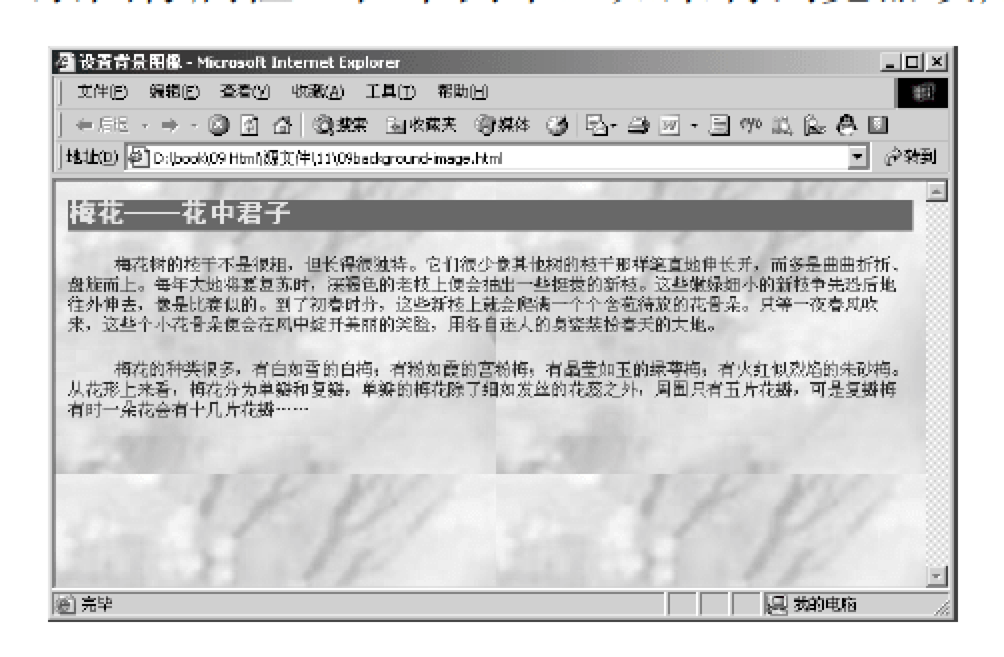


图 11-9　设置背景图像

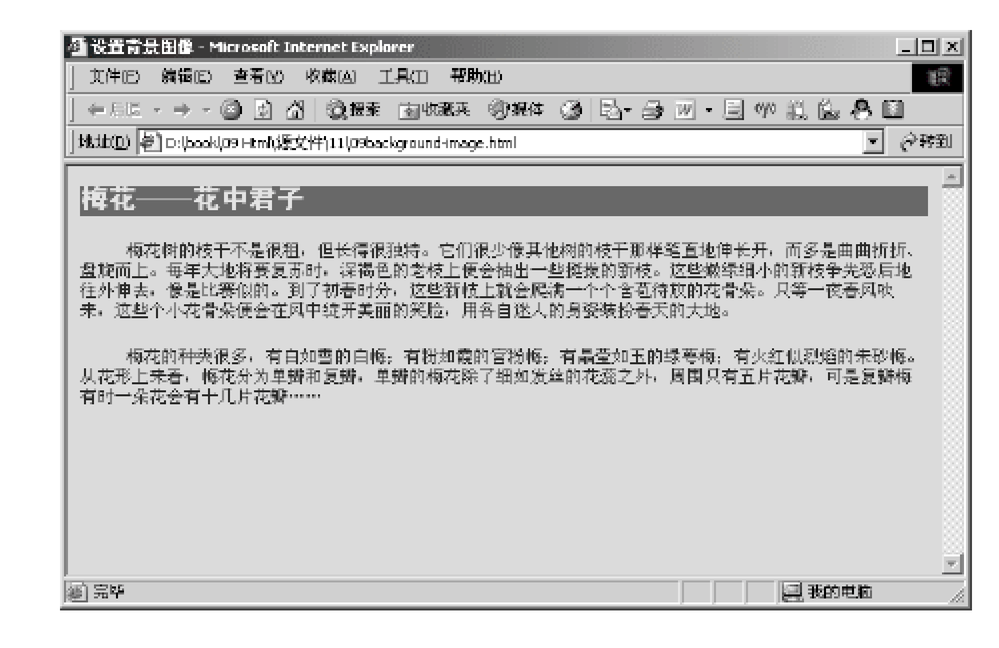


图 11-10　不显示背景图像

为了说明背景图像的其他参数，在后面的实例中，并不会同时设置背景图像和背景颜色。

11.4.4　背景重复——background-repeat

背景重复属性也称为背景图像平铺属性，用来设定对象的背景图像是否以及如何铺排。

语法：background-repeat :取值

取值范围：repeat | no-repeat | repeat-x | repeat-y

说明：repeat 表示背景图像在纵向和横向上平铺；no-repeat 表示背景图像不平铺；repeat-x 表示背景图像只在水平方向上平铺；repeat-y 则表示背景图像只在垂直方向上平铺。

实例代码：

```
<html>
<head>
<title>设置背景重复属性</title>
<style>
<!--
    h2 { font-family:"黑体"; font-size:15pt; color:yellow; background-color:fuchsia }
```

```
        body { background-image:url(pic02.jpg); background-repeat:repeat-x}
        .text { font-family:"宋体"; font-size:10pt; color:#8B0000 }
-->
</style>
</head>
<body>
        <h2>梅花——花中君子</h2>
        <p class=text>     梅花树的枝干不是很粗，但长得很独特。它们很少像其他树的枝干那样笔直地伸长开，而多是曲曲折折、盘旋而上。每年大地将要复苏时，深褐色的老枝上便会抽出一些挺拔的新枝。这些嫩绿细小的新枝争先恐后地往外伸去，像是比赛似的。到了初春时分，这些新枝上就会爬满一个个含苞待放的花骨朵。只等一夜春风吹来，这些个小花骨朵便会在风中绽开美丽的笑脸，用各自迷人的身姿装扮春天的大地。</p>
        <p class=text>     梅花的种类很多，有白如雪的白梅；有粉如霞的宫粉梅；有晶莹如玉的绿萼梅；有火红似烈焰的朱砂梅。从花形上来看，梅花分为单瓣和复瓣，单瓣的梅花除了细如发丝的花蕊之外，周围只有五片花瓣，可是复瓣梅有时一朵花会有十几片花瓣……</p>
</body>
</html>
```

运行代码，看到背景图像只在水平方向平铺的效果，如图 11-11 所示。

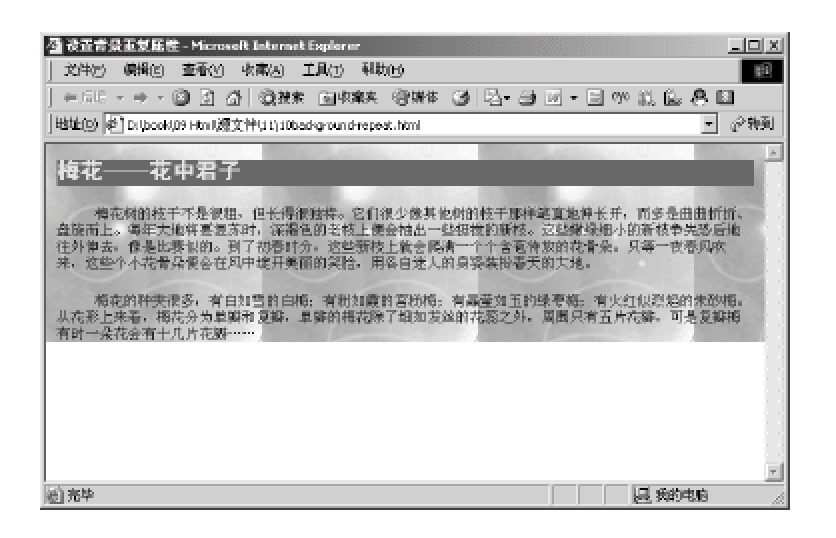


图 11-11　设置背景图像的重复方式

11.4.5　背景附件——background-attachment

背景附件属性用来设置背景图像是随对象内容滚动还是固定的。

语法：background-attachment :取值

取值范围：scroll | fixed

说明：scroll 表示背景图像是随对象内容滚动，是默认选项；fixed 表示背景图像固定在页面上静止不动，只有其他的内容随滚动条滚动。

实例代码：

```
<html>
<head>
<title>设置背景附件</title>
<style>
<!--
        h2 { font-family:"黑体"; font-size:15pt; color:yellow; background-color:fuchsia }
        body { background-image:url(pic02.jpg); background-repeat:no-repeat;
                background-attachment:fixed }
```

```
        .text { font-family:"宋体"; font-size:12pt; color:#8B0000 }
-->
</style>
</head>
<body>
        <h2>梅花——花中君子</h2>
        <p class=text>     梅花树很少像其他树的枝干那样笔直地伸长开，而多是曲曲折折、盘旋而上。每年大地复苏时，深褐色的老枝上便会抽出一些挺拔的新枝。这些嫩绿细小的新枝争先恐后地往外伸去。到了初春时分，这些新枝上就会爬满一个个含苞待放的花骨朵。只等一夜春风吹来，这些个小花骨朵便会在风中绽开美丽的笑脸，用各自迷人的身姿装扮春天的大地。</p>
        <p class=text>     梅花的种类很多，有白如雪的白梅；有粉如霞的宫粉梅；有晶莹如玉的绿萼梅；有火红似烈焰的朱砂梅。从花形上来看，梅花分为单瓣和复瓣，单瓣的梅花除了细如发丝的花蕊之外，周围只有五片花瓣，可是复瓣梅有时一朵花会有十几片花瓣。</p>
        <p class=text>     绿萼梅的花瓣几乎是透明的，像玉琢成的，白中透着几分绿意，显得淡雅高贵。用手摸一摸，像丝绒一般柔滑。这样的梅花很有些冰清玉洁的韵致。朱砂梅则是那样的引人注目，远远望去，一团团、一簇簇的，像跳动的火焰在春风中翩翩起舞……</p>
</body>
</html>
```

运行代码，看到图片开始的位置是页面的左上端，紧贴页面边缘，如图 11-12 所示。拖动滚动条，让页面中的文字向上滚动，发现只有文字向上滚动到显示的页面之外，而背景图像的位置依然在页面的左上端，没有滚动到页面之外，如图 11-13 所示。

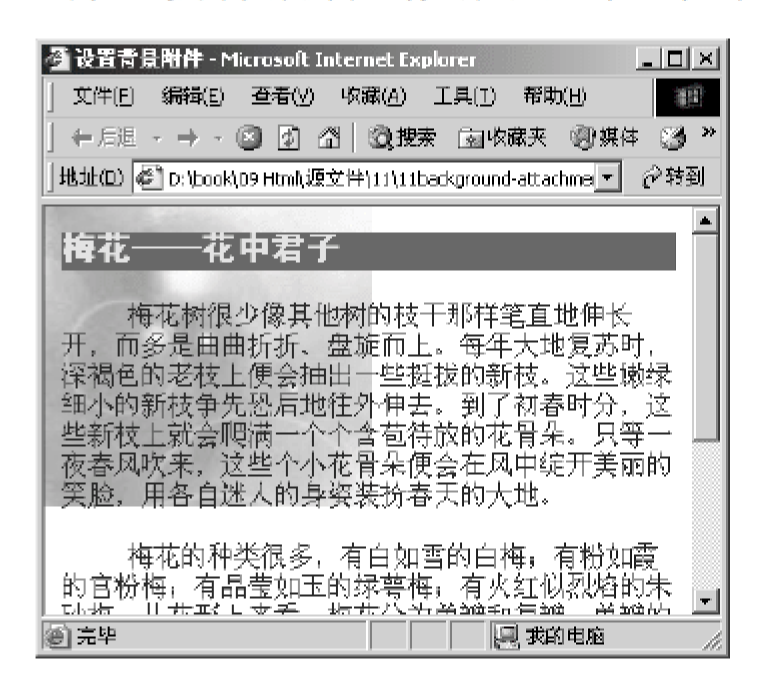


图 11-12　设置背景附件

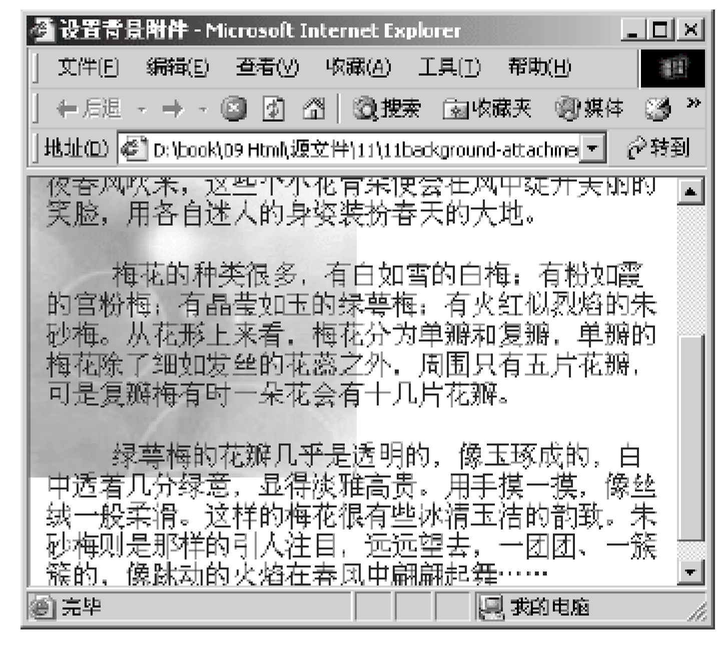


图 11-13　拖动滚动条的效果

11.4.6　背景位置——background-position

背景位置属性用于指定背景图像的最初位置。这个属性只能应用于块级元素和替换元素。替换元素仅指一些已知原有尺寸的元素，在 HTML 中，替换元素包括 img、input、textarea、select 和 object。

语法：background-position: 位置取值

取值范围：[<百分比> | <长度>]{1,2} | [top | center | bottom] || [left | center | right]

说明：该语法中的取值范围包括两种，一种是采用数字，即[<百分比> | <长度>]{1,2}；另一种是关键字描述，即[top | center | bottom] || [left | center | right]，它们的具体含义如下：

- [<百分比> | <长度>]{1,2}：表示使用确切的数字表示图像位置，使用时应首先指定横向位置，接着是纵向位置。例如20% 65%，指定图像会被放在元素的左起20%、上起65%的那点的所在

位置。百分比和长度可以混合使用，设定为负值也是允许的。默认情况下，位置的取值是0% 0%。

- [top | center | bottom] || [left | center | right]：left、center、right是横向的关键字，它们表示在横向上取0%、50%、100%的位置，即采用居左、居中和居右；top、center、bottom是纵向的关键字，它们分别表示在纵向上取0%、50%、100%的位置，即位于顶端、居中和底端。

实例代码：

```
<html>
<head>
<title>设置背景位置</title>
<style>
<!--
    h2 { font-family:"黑体"; font-size:15pt; color:yellow; background-color:fuchsia }
    body { background-image:url(pic02.jpg); background-repeat:no-repeat;
            background-position:left bottom }
    .text { font-family:"宋体"; font-size:10pt; color:#8B0000 }
-->
</style>
</head>
<body>
    <h2>梅花——花中君子</h2>
    <p class=text>     梅花树的枝干不是很粗，但长得很独特。它们很少像其他树的枝干那样笔直地伸长开，而多是曲曲折折、盘旋而上。每年大地将要复苏时，深褐色的老枝上便会抽出一些挺拔的新枝。这些嫩绿细小的新枝争先恐后地往外伸去，像是比赛似的。到了初春时分，这些新枝上就会爬满一个个含苞待放的花骨朵。只等一夜春风吹来，这些个小花骨朵便会在风中绽开美丽的笑脸，用各自迷人的身姿装扮春天的大地。</p>
    <p class=text>     梅花的种类很多，有白如雪的白梅；有粉如霞的宫粉梅；有晶莹如玉的绿萼梅；有火红似烈焰的朱砂梅。从花形上来看，梅花分为单瓣和复瓣，单瓣的梅花除了细如发丝的花蕊之外，周围只有五片花瓣，可是复瓣梅有时一朵花会有十几片花瓣……</p>
</body>
</html>
```

运行这段代码，看到背景图像位于页面的左下角，如图 11-14 所示。

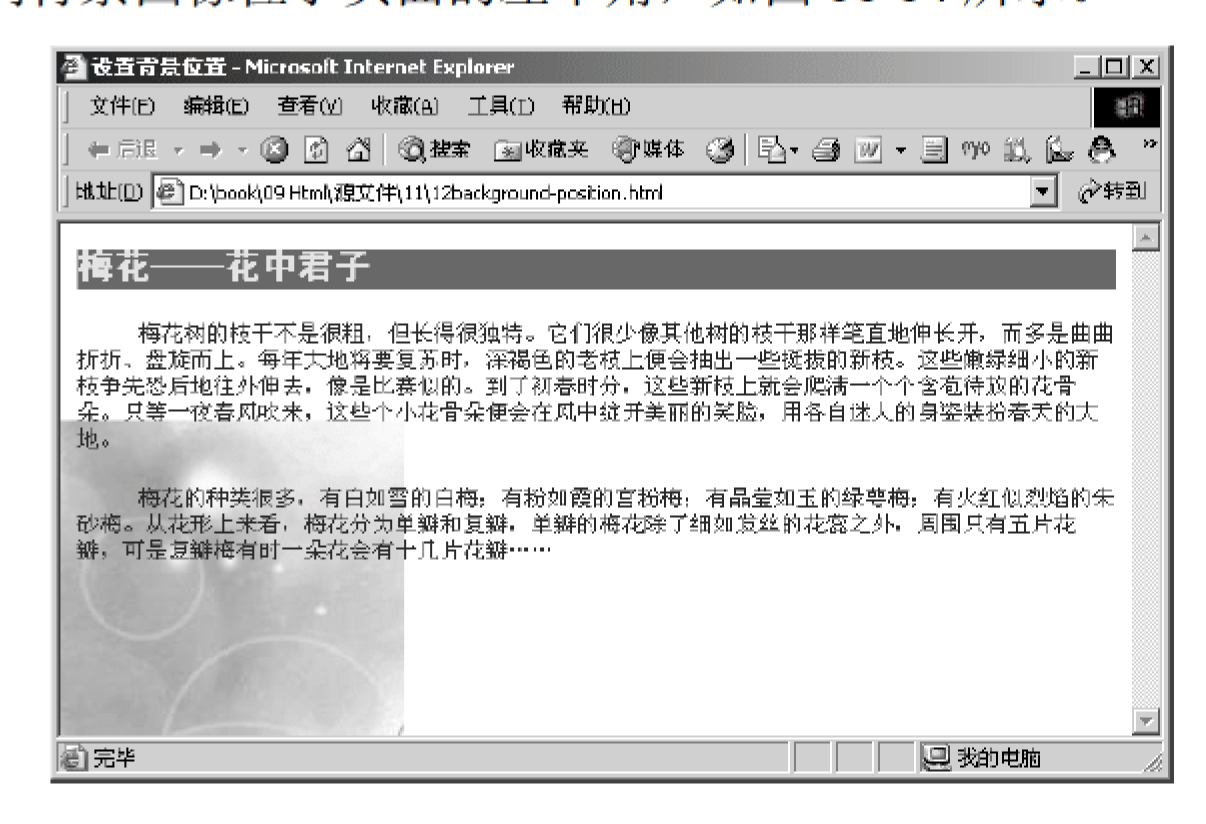


图 11-14　设置背景位置

11.4.7　复合属性：背景——background

与字体 font 属性类似，背景 background 也是复合属性，它是一个更明确的背景关系属性的略写。

语法：background: 取值

说明：在这里，取值范围可以包含背景颜色、背景图像、重复属性、附件属性和背景位置，之间用空格相连。例如，可以设置为 body{ background:#00F url(pic02.jpg) no-repeat}，表示<body>标记内的背景颜色为蓝色 blue，背景图像为 pic02.jpg，图像不重复显示。

实例代码：

```
<html>
<head>
<title>设置背景属性</title>
<style>
<!--
    h2 { font-family:"黑体"; font-size:15pt; color:yellow; background-color:fuchsia }
    body { background: url(pic02.jpg) no-repeat right bottom }
    .text { font-family:"宋体"; font-size:10pt; color:#8B0000 }
-->
</style>
</head>
<body>
    <h2>梅花——花中君子</h2>
    <p class=text>     梅花树的枝干不是很粗，但长得很独特。它们很少像其他树的枝干那样笔直地伸长开，而多是曲曲折折、盘旋而上。每年大地将要复苏时，深褐色的老枝上便会抽出一些挺拔的新枝。这些嫩绿细小的新枝争先恐后地往外伸去，像是比赛似的。到了初春时分，这些新枝上就会爬满一个个含苞待放的花骨朵。只等一夜春风吹来，这些个小花骨朵便会在风中绽开美丽的笑脸，用各自迷人的身姿装扮春天的大地。</p>
    <p class=text>     梅花的种类很多，有白如雪的白梅；有粉如霞的宫粉梅；有晶莹如玉的绿萼梅；有火红似烈焰的朱砂梅。从花形上来看，梅花分为单瓣和复瓣，单瓣的梅花除了细如发丝的花蕊之外，周围只有五片花瓣，可是复瓣梅有时一朵花会有十几片花瓣……</p>
</body>
</html>
```

运行代码，看到成功实现了对背景的各种参数的设置，如图 11-15 所示。

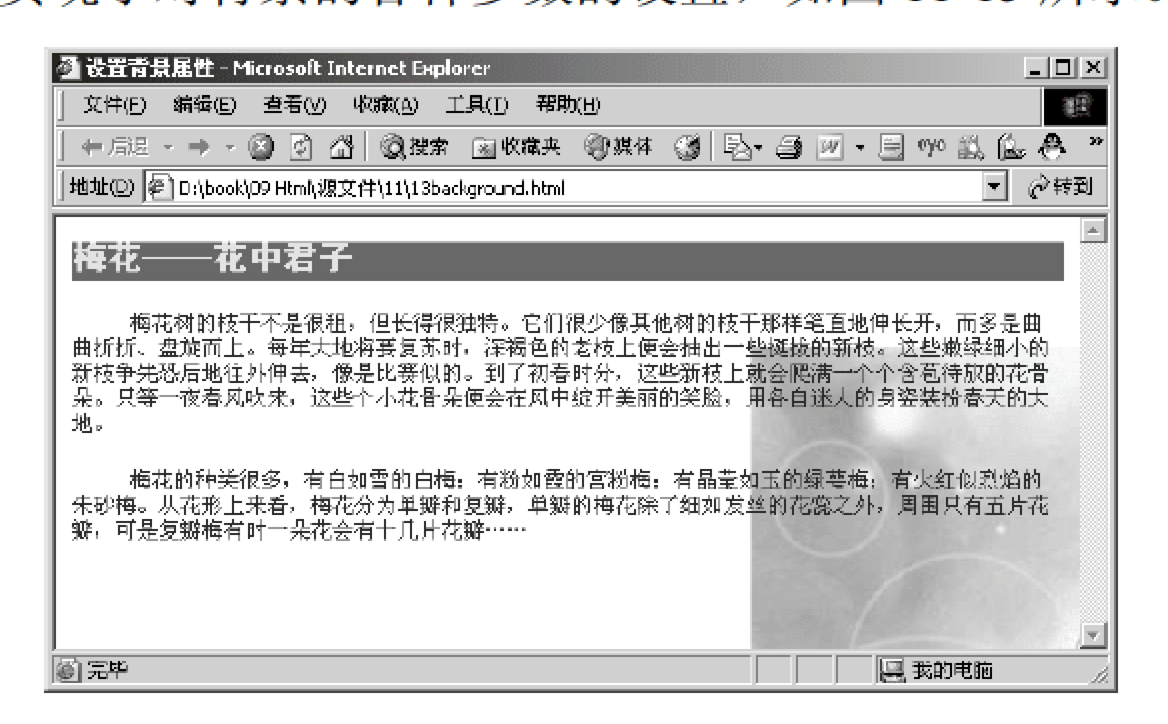


图 11-15　设置背景属性

11.5　文本属性

在 CSS 中，文本属性主要包括单词间隔、字符间隔、文字修饰、纵向排列、文本转换、文本排列、文本缩进、行高等几种属性。

11.5.1　单词间隔——word-spacing

单词间隔用来定义附加在单词之间的间隔的数量，但其取值必须符合长度格式。单词间隔的设置多用于英文文本中。

语法：word-spacing:取值

取值范围：normal | <长度>

说明：normal 是指正常的间隔，是默认选项；长度是设定单词间隔的数值及单位，可以使用负值。

实例代码：

```
<html>
<head>
<title>设置单词间隔</title>
<style>
<!--
	h2 {font-size:18pt; word-spacing:8px}
	.text { font-size:12pt; word-spacing: normal}
	.newtext { font-size:12pt;   word-spacing: 7px}
-->
</style>
</head>
<body>
	<h2> contrast </h2>
	<p class=text>The craftsman, Arnold, came from a family of carpenters. </p>
	<p class=newtext>The craftsman, Arnold, came from a family of carpenters. </p>
</body>
</html>
```

运行这段代码，可以看到两段内容相同的文字由于设置了不同的单词间隔而显示出不同的效果，如图 11-16 所示。

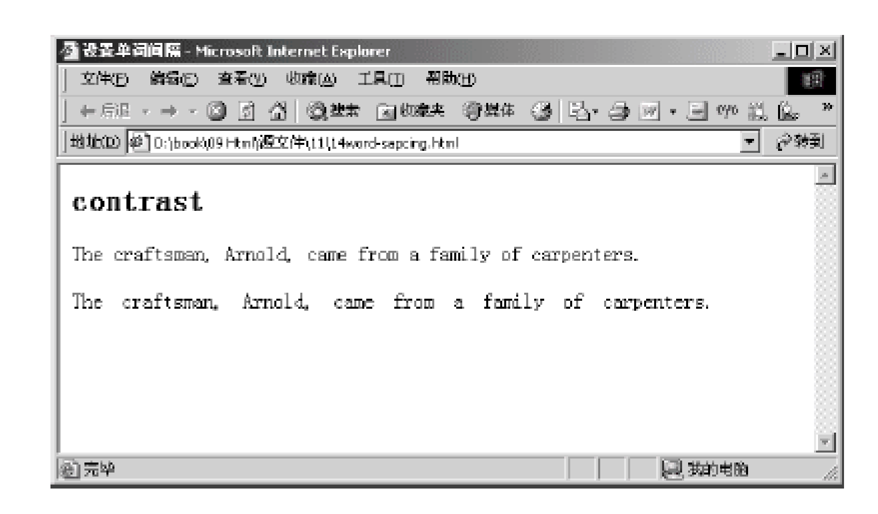


图 11-16　设置单词间隔

11.5.2　字符间隔——letter-spacing

字符间隔和单词间隔类似，不同的是字符间隔用于设置字符的间隔数。

语法：letter-spacing:取值

取值范围：normal | <长度>

实例代码：

```
<html>
<head>
<title>设置字符间隔</title>
<style>
<!--
	h2 {font-family:黑体; font-size:16pt; letter-spacing:10px}
	.text {font-size:10pt; letter-spacing: normal}
	.newtext {font-size:10pt; letter-spacing: 4px}
-->
</style>
</head>
<body>
	<h2>木兰从军</h2>
	<p class=text>     北魏末年，柔然、契丹等少数民族日渐强大，他们经常派兵侵扰中原地区，抢劫财物。北魏朝廷为了对付他们，常常大量征兵，加强北部边境的驻防。木兰从小跟着父亲读书写字。她还喜欢骑马射箭，练得一身好武艺。</p>
	<p class=newtext>    有一天，衙门里的差役送来了征兵的通知，要征木兰的父亲去当兵。但父亲年纪老迈，又怎能参军打仗呢？木兰没有哥哥，弟弟又太小，她不忍心让年老的父亲去受苦，于是决定女扮男装，代父从军。木兰父母虽不舍得女儿出征，但又无他法，只好同意她去了。木兰随着队伍，到了北方边境。她担心自己女扮男装的秘密被人发现，故此处处加倍小心。白天行军，木兰紧紧地跟上队伍，从不敢掉队。夜晚宿营，她从来不敢脱衣服。作战的时候，她凭着一身好武艺，总是冲杀在前……</p>
</body>
</html>
```

运行代码，可以看到两段文字设置了不同字符间隔的效果，如图 11-17 所示。

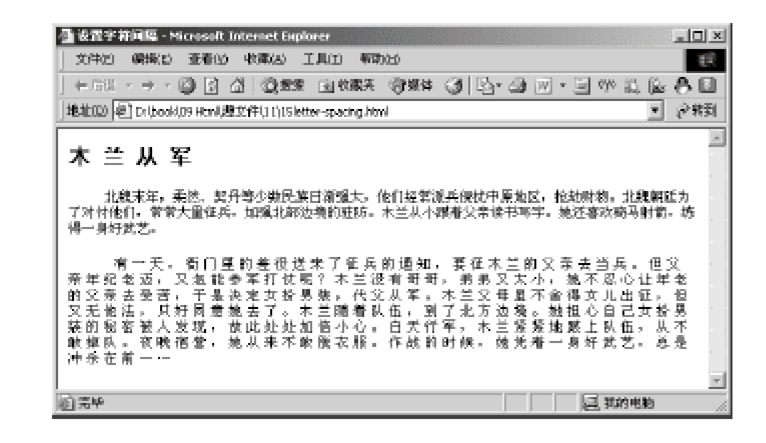


图 11-17　设置字符间隔

11.5.3　文字修饰——text-decoration

文字修饰属性主要是用于对文本进行修饰，如设置下划线、删除线等。

语法：text-decoration:修饰值

取值范围：none | [underline || overline || line-through || blink]

说明：none 表示不对文本进行修饰，这是默认属性值；underline 表示对文字添加下划线；overline 表示对文本添加上划线；line-through 表示对文本添加删除线；blink 则表示文字闪烁效果，这一属性值只有在 Netscape 浏览器中才能正常显示。

实例代码：

```
<html>
<head>
<title>文字修饰</title>
<style>
<!--
     h2 {font-family:黑体; font-size:16pt; letter-spacing:10px; text-decoration: none}
     .text {font-size:10pt; text-decoration: underline}
     .newtext {font-size:10pt; text-decoration: line-through}
-->
</style>
</head>
<body>
     <h2>木兰从军</h2>
     <p class=text>北魏末年，柔然、契丹等少数民族日渐强大，他们经常派兵侵扰中原地区，抢劫财物。北魏朝廷为了对付他们，常常大量征兵，加强北部边境的驻防。木兰从小跟着父亲读书写字。她还喜欢骑马射箭，练得一身好武艺。</p>
     <p class=newtext>有一天，衙门里的差役送来了征兵的通知，要征木兰的父亲去当兵。但父亲年纪老迈，又怎能参军打仗呢？木兰没有哥哥，弟弟又太小，她不忍心让年老的父亲去受苦，于是决定女扮男装，代父从军。木兰父母虽不舍得女儿出征，但又无他法，只好同意她去了。木兰随着队伍，到了北方边境。她担心自己女扮男装的秘密被人发现，故此处处加倍小心。白天行军，木兰紧紧地跟上队伍，从不敢掉队。夜晚宿营，她从来不敢脱衣服。作战的时候，她凭着一身好武艺，总是冲杀在前……</p>
</body>
</html>
```

运行这段代码，看到第一段的段落文字使用了下划线，第二段文字则使用了删除线，效果如图 11-18 所示。

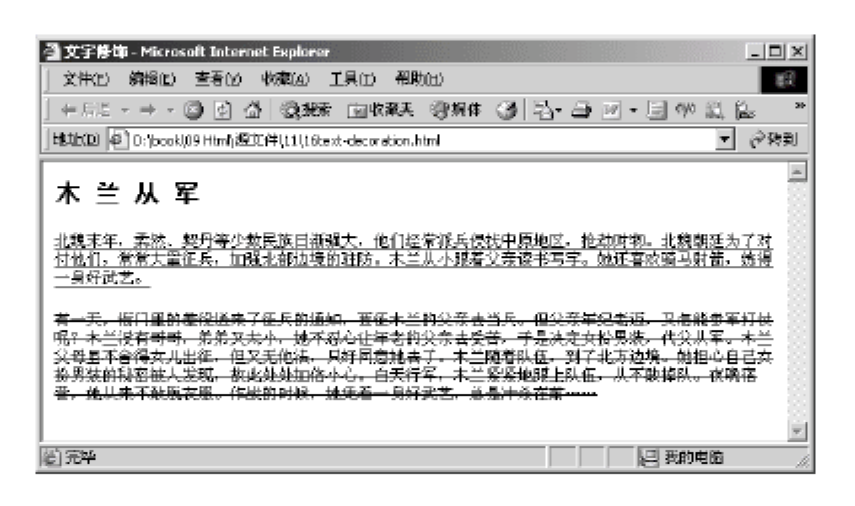


图 11-18　设置文字修饰属性

11.5.4　纵向排列——vertical-align

纵向排列属性也称为垂直对齐方式，它可以设置一个内部元素的纵向位置，相对于它的上级元素或相对于元素行。内部元素是没有行在其前和后断开的元素，例如，在 HTML 中的 A 和 IMG。它主要

用于对图像的纵向排列。

语法：vertical-align:排列取值

取值范围：baseline | sub | super | top | text-top | middle | bottom | text-bottom | <百分比>

说明：baseline 使元素和上级元素的基线对齐；sub 为下标；super 为上标；top 使元素和行中最多的元素向上对齐；text-top 使元素和上级元素的字体向上对齐；middle 是纵向对齐元素基线加上上级元素的 x 高度的一半的中点，其中 x 高度是字母“ x ”的高度；text-bottom 使元素和上级元素的字体向下对齐。

影响相对于元素行的位置的关键字有 top 和 bottom。其中，top 使元素和行中最高的元素向上对齐；bottom 使元素和行中最低的元素向下对齐。

百分比是一个相对于元素行高属性的百分比，它会在上级基线上增高元素基线的指定数量。这里允许使用负值，负值表示减少相应的数量。

实例代码：

```
<html>
<head>
<title>纵向排列</title>
<style>
<!--
    body{font-size:12pt;}
    img{ vertical-align:-100%}
    .exponent{font-size:9pt; vertical-align: super }
-->
</style>
</head>
<body>
    在方程式中的指数部分应用上标的效果：<br>
    X<font class=exponent>3</font>-3=0<br><br>
    这是一幅图像：<img src="pic03.jpg">
</body>
</html>
```

运行代码，可以看到效果如图 11-19 所示。



图 11-19　设置文本的纵向排列

11.5.5　文本转换——text-transform

文本转换属性仅被用于表达某种格式的要求，确切地说，是用来转换英文文字的大小写的。

语法：text-transform: 转换值

取值范围：none | capitalize | uppercase | lowercase

说明：文本转换属性允许通过 4 个属性中的一个来转换文本。其中，none 表示使用原始值；capitalize 使每个字的第一个字母大写；uppercase 使每个字的所有字母大写；lowercase 则使每个字的所有字母小写。

实例代码：

```
<html>
<head>
```

```
<title>设置英文的大小写</title>
<style>
<!--
     h2{ font-size:14pt}
     .textn{ font-size:12pt; text-transform: none}
     .textc{ font-size:12pt; text-transform: capitalize }
     .textu{ font-size:12pt; text-transform: uppercase }
     .textl{ font-size:12pt; text-transform: lowercase }
-->
</style>
</head>
<body>
     <h2>下面是相同的一句话设置不同的转换值效果：</h2>
     <p class=textn>The craftsman, Arnold, came from a family of carpenters. </p>
     <p class=textc>The craftsman, Arnold, came from a family of carpenters. </p>
     <p class=textu>The craftsman, Arnold, came from a family of carpenters. </p>
     <p class=textl>The craftsman, Arnold, came from a family of carpenters. </p>
</body>
</html>
```

运行这段代码，可以看到设置了不同转换值的段落文字显示方式也不同，如图 11-20 所示。

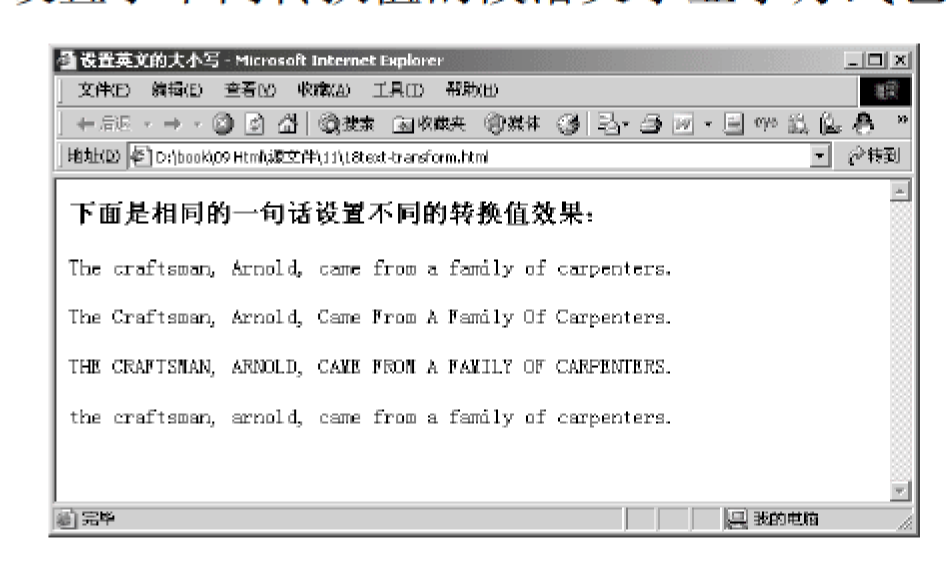


图 11-20　设置不同的转换值效果

11.5.6　文本排列——text-align

文本排列属性能够使元素文本得到排列。这个属性的功能类似于 HTML 的段、标题和部分的 align 属性。

语法：text-align:排列值

取值范围：left | right | center | justify

说明：在该语法中，可以选择 4 个对齐方式中的一个。其中，left 为左对齐；right 为右对齐；center 为居中对齐；justify 为两端对齐。

实例代码：

```
<html>
<head>
<title>文本排列</title>
<style>
<!--
```

```
        h2 {font-family:黑体; font-size:16pt; letter-spacing:10px; text-align: center}
        .text {font-size:10pt; text-align: left}
        .newtext {font-size:10pt; text-align: right}
-->
</style>
</head>
<body>
        <h2>木兰从军</h2>
        <p class=text>北魏末年，柔然、契丹等少数民族日渐强大，他们经常派兵侵扰中原地区，抢劫财物。北魏朝廷为了对付他们，常常大量征兵，加强北部边境的驻防。木兰从小跟着父亲读书写字。她还喜欢骑马射箭，练得一身好武艺。</p>
        <p class=newtext>有一天，衙门里的差役送来了征兵的通知，要征木兰的父亲去当兵。但父亲年纪老迈，又怎能参军打仗呢？木兰没有哥哥，弟弟又太小，她不忍心让年老的父亲去受苦，于是决定女扮男装，代父从军。木兰父母虽不舍得女儿出征，但又无他法，只好同意她去了。木兰随着队伍，到了北方边境。她担心自己女扮男装的秘密被人发现，故此处处加倍小心。白天行军，木兰紧紧地跟上队伍，从不敢掉队。夜晚宿营，她从来不敢脱衣服。作战的时候，她凭着一身好武艺，总是冲杀在前……</p>
</body>
</html>
```

运行这段代码，可以看到如图 11-21 所示的效果。

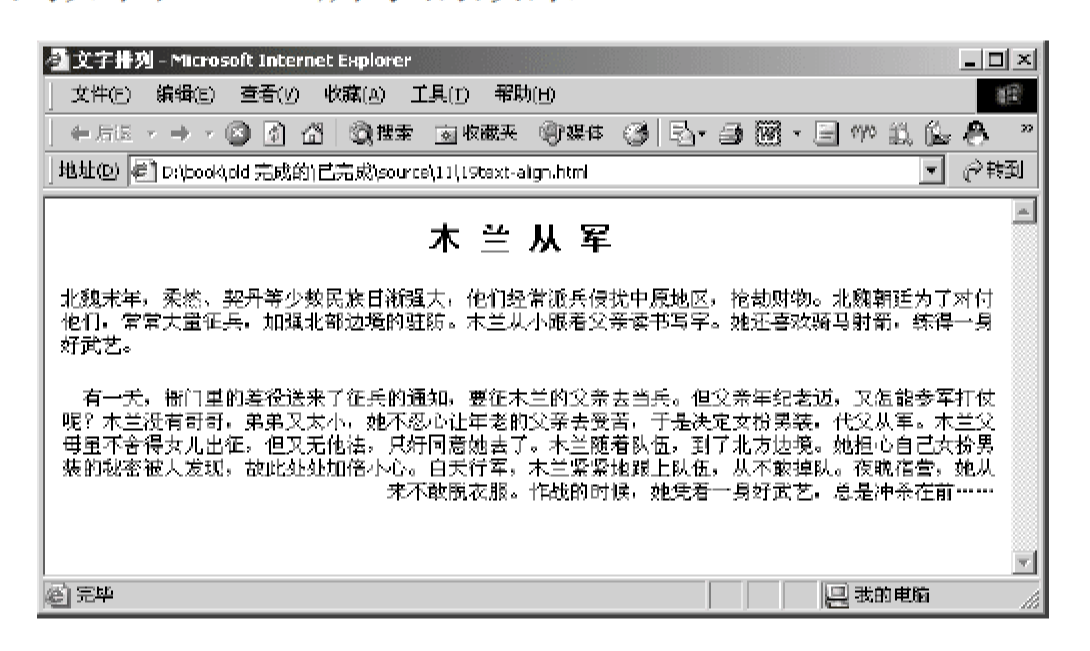


图 11-21　设置文本对齐方式

11.5.7　文本缩进——text-indent

文本缩进属性用于定义 HTML 中级元素（如 p、h1 等）的第一行可以接受的缩进数量，常用于设置段落的首行缩进。

语法：text-indent:缩进值

说明：文本的缩进值必须是一个长度或一个百分比。若设定为百分比，则以上级元素的宽度而定。

实例代码：

```
<html>
<head>
<title>设置文本缩进</title>
<style>
<!--
        h2 {font-family:黑体; font-size:16pt}
        .text {font-size:10pt; text-indent:50%}
```

```
        .newtext {font-size:10pt; text-indent:20pt}
-->
</style>
</head>
<body>
        <h2>木兰从军</h2>
        <p class=text>北魏末年，柔然、契丹等少数民族日渐强大，他们经常派兵侵扰中原地区，抢劫财物。北魏朝廷为了对付他们，常常大量征兵，加强北部边境的驻防。木兰从小跟着父亲读书写字。她还喜欢骑马射箭，练得一身好武艺。</p>
        <p class=newtext>有一天，衙门里的差役送来了征兵的通知，要征木兰的父亲去当兵。但父亲年纪老迈，又怎能参军打仗呢？木兰没有哥哥，弟弟又太小，她不忍心让年老的父亲去受苦，于是决定女扮男装，代父从军。木兰父母虽不舍得女儿出征，但又无他法，只好同意她去了。木兰随着队伍，到了北方边境。她担心自己女扮男装的秘密被人发现，故此处处加倍小心。白天行军，木兰紧紧地跟上队伍，从不敢掉队。夜晚宿营，她从来不敢脱衣服。作战的时候，她凭着一身好武艺，总是冲杀在前……</p>
</body>
</html>
```

运行程序，效果如图 11-22 所示。其中第一段设定了首行缩进为 50%，段落的上一级元素是<body>，因此首行缩进文本主体宽度的一半；第二段设定了首行缩进为 20pt。

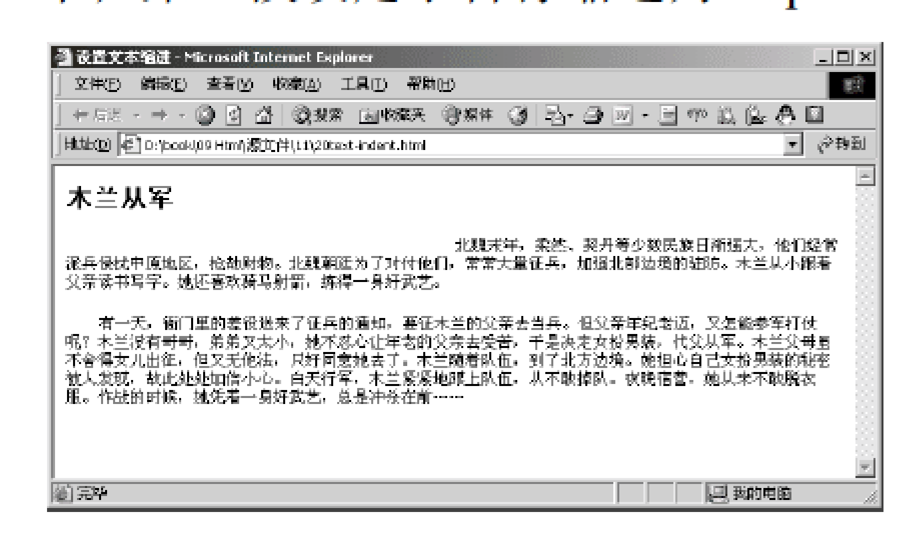


图 11-22　设置文本的首行缩进

11.5.8　文本行高——line-height

行高属性用于控制文本基线之间的间隔值。

语法：line-height:行高值

取值范围：normal | <数字> | <长度> | <百分比>

说明：normal 表示默认的行高，一般由字体大小的属性自动产生；值为数字时，行高由元素字体大小的量与该数字相乘所得；长度属性则是直接使用数字和单位设置行高；值为百分比时，表示相对于元素字体大小的比例，不允许使用负值。

实例代码：

```
<html>
<head>
<title>设置文本的行高</title>
<style>
<!--
        h2 {font-family:黑体; font-size:16pt; line-height:1.2}
        .text {font-size:10pt; text-indent:10%; line-height: 12pt}
```

```
	.newtext {font-size:10pt; text-indent:20pt; line-height: 200%}
-->
</style>
</head>
<body>
	<h2>木兰从军</h2>
	<p class=text>北魏末年，柔然、契丹等少数民族日渐强大，他们经常派兵侵扰中原地区，抢劫财物。北魏朝廷为了对付他们，常常大量征兵，加强北部边境的驻防。木兰从小跟着父亲读书写字。她还喜欢骑马射箭，练得一身好武艺。</p>
	<p class=newtext>有一天，衙门里的差役送来了征兵的通知，要征木兰的父亲去当兵。但父亲年纪老迈，又怎能参军打仗呢？木兰没有哥哥，弟弟又太小，她不忍心让年老的父亲去受苦，于是决定女扮男装，代父从军。木兰父母虽不舍得女儿出征，但又无他法，只好同意她去了。木兰随着队伍，到了北方边境。她担心自己女扮男装的秘密被人发现，故此处处加倍小心。白天行军，木兰紧紧地跟上队伍，从不敢掉队。夜晚宿营，她从来不敢脱衣服。作战的时候，她凭着一身好武艺，总是冲杀在前……</p>
</body>
</html>
```

运行代码，可以看到两个段落设置不同行高的效果，如图 11-23 所示。

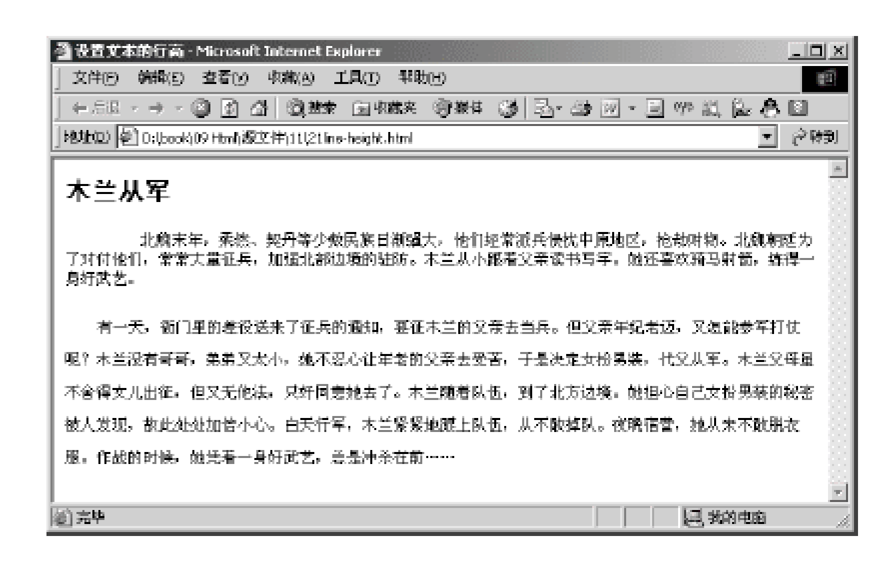


图 11-23　设置行高

11.5.9　处理空白——white-space

white-space 属性用于设置页面对象内空白（包括空格和换行等）的处理方式。默认情况下，HTML 中的连续多个空格会被合并成一个，而使用这一属性可以设置成其他的处理方式。

语法：white-space :值

取值范围：normal | pre | nowrap

说明：white-space 只能取 3 个值中的一个，其中 normal 是默认属性，即将连续的多个空格合并；pre 会导致源中的空格和换行符被保留，但这一选项只有在 Internet Explorer 6 中才能正确显示；nowrap 则表示强制在同一行内显示所有文本，直到文本结束或者遇到
对象。

实例代码：

```
<html>
<head>
<title>设置空格处理方式</title>
<style>
<!--
	h2 {font-family:黑体; font-size:16pt }
```

```
        .text {font-size:10pt; text-indent:20pt; white-space: nowrap}
        .newtext {font-size:10pt; text-indent:20pt}
-->
</style>
</head>
<body>
        <h2>木兰从军</h2>
        <p class=text>北魏末年，柔然、契丹等少数民族日渐强大，他们经常派兵侵扰中原地区，抢劫财物。北魏朝廷为了对付他们，常常大量征兵，加强北部边境的驻防。木兰从小跟着父亲读书写字。她还喜欢骑马射箭，练得一身好武艺。</p>
        <p class=newtext>有一天，衙门里的差役送来了征兵的通知，要征木兰的父亲去当兵。但父亲年纪老迈，又怎能参军打仗呢？木兰没有哥哥，弟弟又太小，她不忍心让年老的父亲去受苦，于是决定女扮男装，代父从军。木兰父母虽不舍得女儿出征，但又无他法，只好同意她去了。木兰随着队伍，到了北方边境。她担心自己女扮男装的秘密被人发现，故此处处加倍小心。白天行军，木兰紧紧地跟上队伍，从不敢掉队。夜晚宿营，她从来不敢脱衣服。作战的时候，她凭着一身好武艺，总是冲杀在前……</p>
</body>
</html>
```

运行程序，效果如图 11-24 所示。页面第一个段落的文字被强迫在同一行内显示。

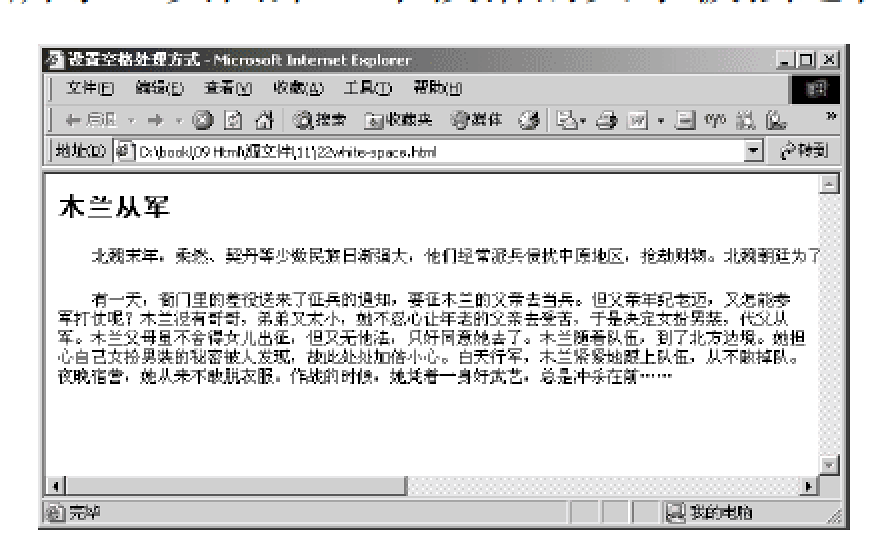


图 11-24　设置空白属性

11.5.10　文本反排——unicode-bidi 与 direction

unicode-bidi 属性用于设置或获取关于双向法则的嵌入级别，通俗一些地说，就是用于同一个页面里存在从不同方向读进的文本的显示，它一般与 direction 属性一起使用，direction 属性用来设置对象的阅读顺序。

1．unicode-bidi 属性

语法：unicode-bidi :normal | bidi-override | embed

说明：normal 是系统的默认值，表示对象不打开附加的嵌入层；bidi-override 表示严格按照 direction 属性的值重排序，忽略隐式双向运算规则；embed 表示对象打开附加的嵌入层，将 direction 属性的值指定给嵌入层，在对象内部进行隐式重排序。如果想要在内联文本中应用 direction 属性，必须设定该属性的值为 embed 或 bidi-override。

注意： Unicode 双向运算法则是自动翻转嵌入字符顺序依照它们固有的流动方向。例如，英文文档的默认书写方向是从左到右，假如其中包含的部分其他语种的字符其书写方向是从右到左，双向运算法则就可以用来代理用户正确地反转其流动方向。

2．direction 属性

语法：direction : ltr | rtl | inherit

说明：direciton 属性的值中，ltr 表示从左到右的阅读顺序；rtl 表示从右到左的阅读顺序；inherit 则表示文本流的值不可继承。

下面通过实例说明这两个属性结合使用的效果，实例代码如下：

```
<html>
<head>
<title>文本反排</title>
<style>
<!--
      h2 {font-family:黑体; font-size:16pt }
      .text {font-size:10pt; text-indent:20pt;direction: rtl; unicode-bidi: bidi-override;}
      .newtext {font-size:10pt; text-indent:20pt}
-->
</style>
</head>
<body>
      <h2>木兰从军</h2>
      <p class=text>北魏末年，柔然、契丹等少数民族日渐强大，他们经常派兵侵扰中原地区，抢劫财物。北魏朝廷为了对付他们，常常大量征兵，加强北部边境的驻防。木兰从小跟着父亲读书写字。她还喜欢骑马射箭，练得一身好武艺。</p>
      <p class=newtext>有一天，衙门里的差役送来了征兵的通知，要征木兰的父亲去当兵。但父亲年纪老迈，又怎能参军打仗呢？木兰没有哥哥，弟弟又太小，她不忍心让年老的父亲去受苦，于是决定女扮男装，代父从军。木兰父母虽不舍得女儿出征，但又无他法，只好同意她去了。木兰随着队伍，到了北方边境。她担心自己女扮男装的秘密被人发现，故此处处加倍小心。白天行军，木兰紧紧地跟上队伍，从不敢掉队。夜晚宿营，她从来不敢脱衣服。作战的时候，她凭着一身好武艺，总是冲杀在前……</p>
</body>
</html>
```

运行程序，效果如图 11-25 所示。在页面中的第一段文字，设置了文本从右到左的阅读顺序，从而实现了文本的反排。

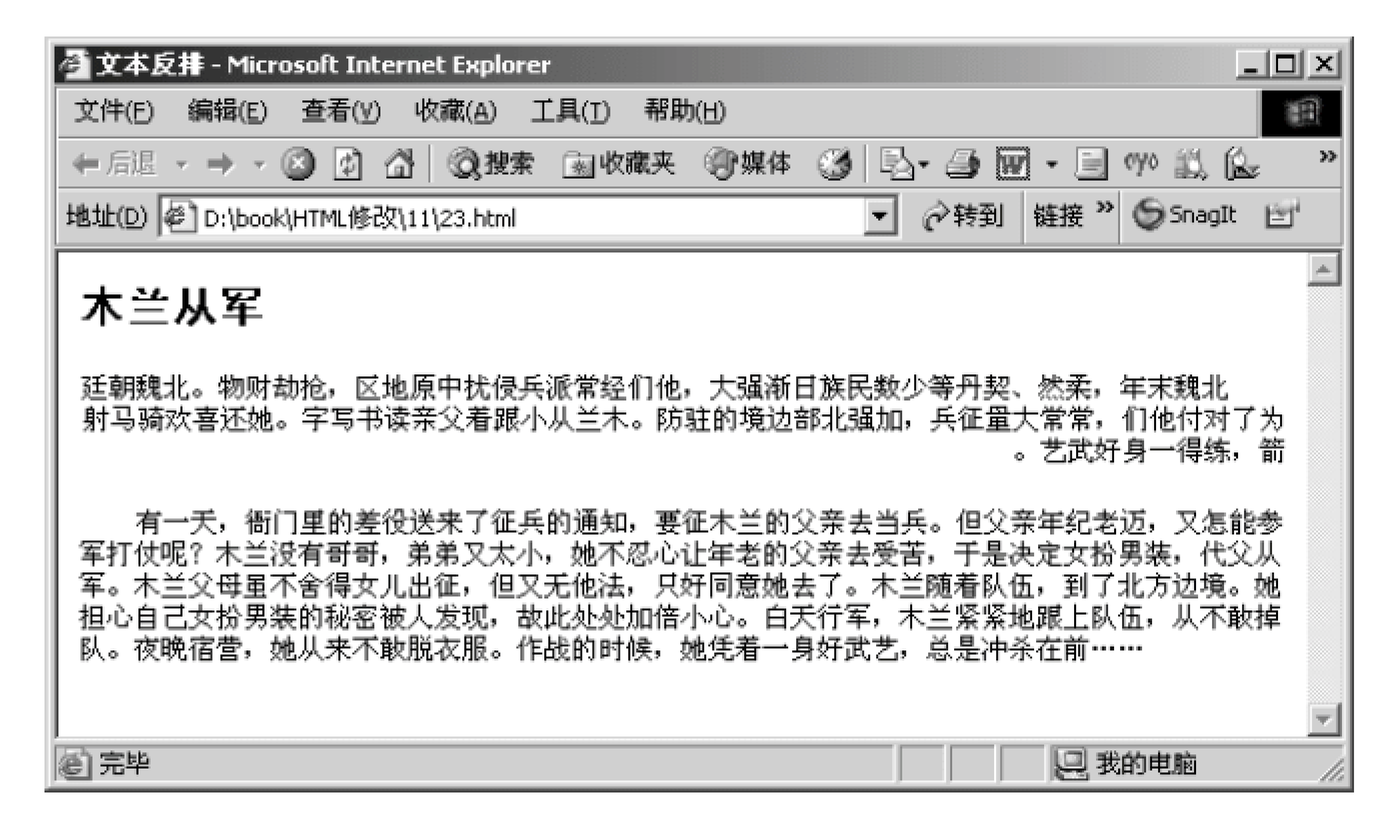


图 11-25　设置文本的反排

11.6　边距与填充属性

边距属性用于设置元素周围的边界宽度，主要包括上下左右 4 个边界的距离设置。填充属性也称为补白属性，用于设置边框和元素内容之间的间隔数，同样包括上、下、左、右 4 个方向的填充值。

11.6.1　顶端边距——margin-top

顶端边距属性也称上边距，其用一个指定的长度或百分比值来设置元素的上边界。

语法：margin-top:边距值

取值范围：长度值 | 百分比 | auto

说明：长度值相当于设置顶端的绝对边距值，包括数字和单位；百分比值则是设置相对于上级元素的宽度的百分比，允许使用负值；auto 是自动取边距值，即取元素的默认值。

实例代码：

```
<html>
<head>
<title>设置顶端边距</title>
<style>
<!--
    h2 {font-family:黑体; font-size:16pt; margin-top:15pt}
    p {font-size:11pt; text-indent:20pt}
    img { margin-top:25pt}
-->
</style>
</head>
<body>
    <h2>城市猎人</h2>
    <p><img src="pic04.jpg" align="left">日本著名写实漫画家北条司名作，以动感与质感完美融合的画风，描绘了一个极富现代感的侦探，这部故事的主角寒羽良【港译“孟波”】，是以《猫眼》中的一个叫神谷的配角为原型的，神谷是一个有正义感的神偷，而寒羽良则成为一个见义勇为、武艺超群，而且足智多谋的职业侦探，这两个人有一个共同的特点，就是十分斗趣，总爱在女孩子面前开一些过火的玩笑。</p>
</body>
</html>
```

运行这段代码，看到<h2>标记内的文字以及图像被设置了不同的上边距，如图 11-26 所示。

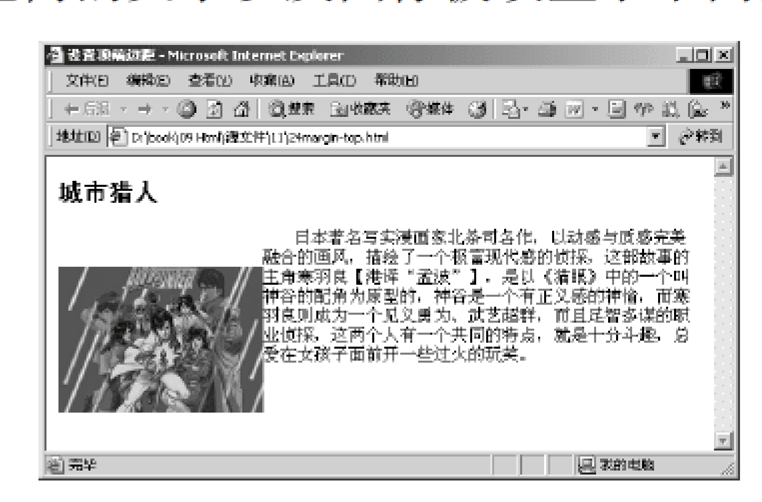


图 11-26　设置上边距

11.6.2　其他边距——margin-bottom、margin-left、margin-right

底端边距用于设置元素下方的边距值；左侧边距和右侧边距则分别用于设置元素左右两侧的边距值。其语法和使用方法同顶端边距类似，下面用一个实例说明底端边距、左侧边距和右侧边距的效果。

实例代码：

```
<html>
<head>
<title>设置其他边距属性</title>
<style>
<!--
    h2 {font-family:黑体; font-size:16pt}
    p {font-size:11pt; text-indent:20pt}
    img { margin-top:10pt; margin-bottom:10pt; margin-left:25pt; margin-right:25pt}
-->
</style>
</head>
<body>
    <h2>城市猎人</h2>
    <p><img src="pic04.jpg" align="left">日本著名写实漫画家北条司名作，以动感与质感完美融合的画风，描绘了一个极富现代感的侦探，这部故事的主角寒羽良【港译“孟波”】，是以《猫眼》中的一个叫神谷的配角为原型的，神谷是一个有正义感的神偷，而寒羽良则成为一个见义勇为、武艺超群，而且足智多谋的职业侦探，这两个人有一个共同的特点，就是十分斗趣，总爱在女孩子面前开一些过火的玩笑。</p>
</body>
</html>
```

运行程序，效果如图 11-27 所示。

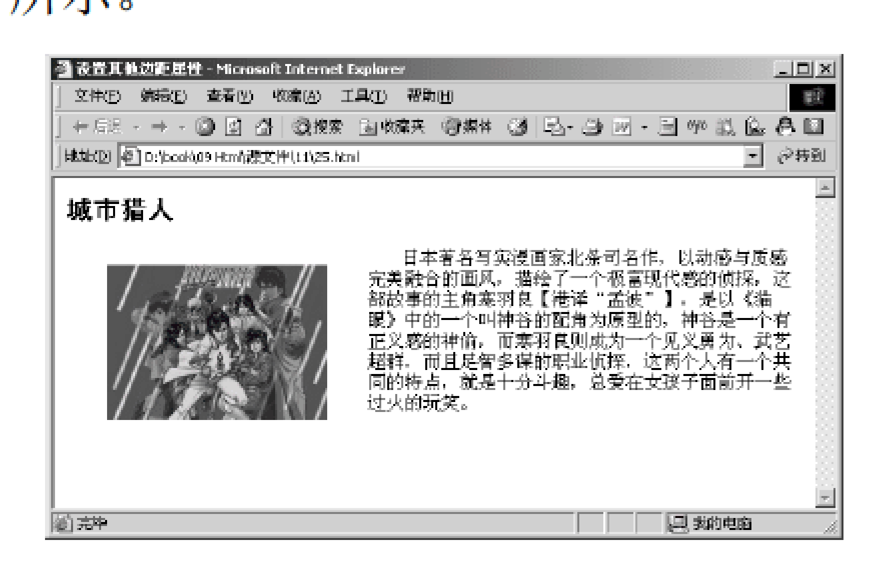


图 11-27　设置图像的上、下、左、右边距

11.6.3　复合属性：边距——margin

与其他属性类似，边距属性是用于对 4 个边距设置的略写。

语法：margin: 长度值 | 百分比 | auto

说明：在这里，margin 的值可以取 1 到 4 个，如果只设置了 1 个值，则应用于所有的 4 个边界；如果设置了两个或 3 个值，则省略的值与对边相等；如果设置了 4 个值，则按照上、右、下、左的顺序分别对应其边距。

实例代码：

```
<html>
<head>
<title>设置边距属性</title>
<style>
<!--
	h2 {font-family:黑体; font-size:16pt}
	p {font-size:11pt; text-indent:20pt}
	img { margin: 8pt 30pt 10pt}
-->
</style>
</head>
<body>
	<h2>城市猎人</h2>
	<p><img src="pic04.jpg" align="left">日本著名写实漫画家北条司名作，以动感与质感完美融合的画风，描绘了一个极富现代感的侦探，这部故事的主角寒羽良【港译“孟波”】，是以《猫眼》中的一个叫神谷的配角为原型的，神谷是一个有正义感的神偷，而寒羽良则成为一个见义勇为、武艺超群，而且足智多谋的职业侦探，这两个人有一个共同的特点，就是十分斗趣，总爱在女孩子面前开一些过火的玩笑。</p>
</body>
</html>
```

运行程序，效果如图 11-28 所示。其中上边距是 8pt，下边距是 10pt，右边距是 30pt，左边距未设置，采用与右边距相等的值，即同样设置为 30pt。

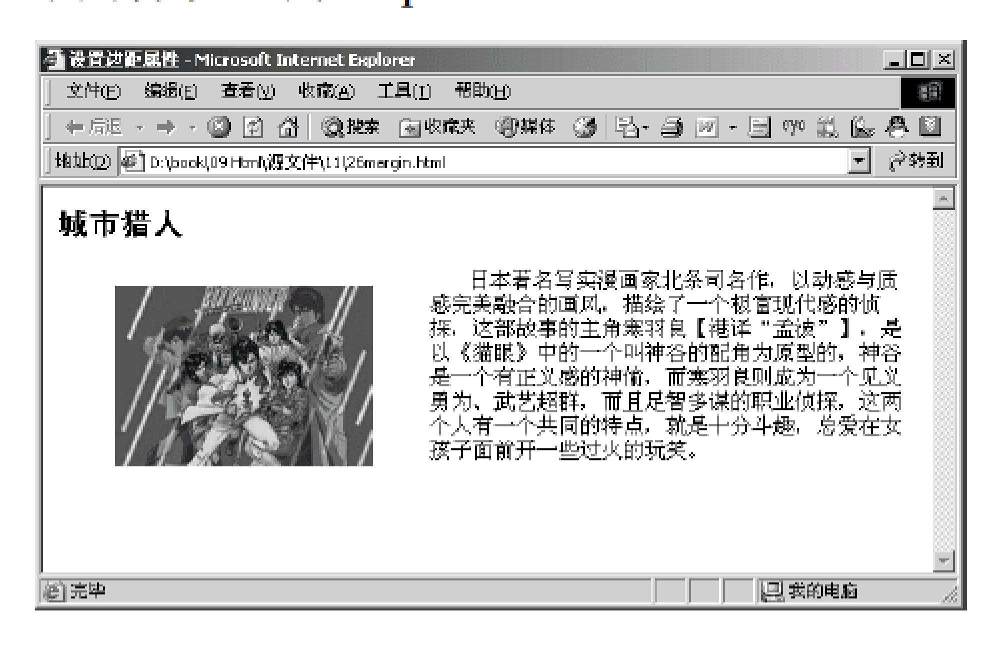


图 11-28　设置边距属性

11.6.4　顶端填充——padding-top

顶端填充属性也称为上补白，即上边框和选择符内容之间的间隔数。

语法：padding-top: 间隔值

说明：间隔值可以设置为长度值或百分比。其中，百分比不能使用负值。

在前面实例的基础上设置 body 的上补白值，其实例代码如下：

```
<html>
<head>
<title>设置顶端填充属性</title>
<style>
```

```
<!--
      Body{padding-top:26pt}
      h2 {font-family:黑体; font-size:16pt}
      p {font-size:11pt; text-indent:20pt}
-->
</style>
</head>
<body>
      <h2>城市猎人</h2>
      <p><img src="pic04.jpg" align="left">日本著名写实漫画家北条司名作，以动感与质感完美融合的画风，描绘了一个极富现代感的侦探，这部故事的主角寒羽良【港译“孟波”】，是以《猫眼》中的一个叫神谷的配角为原型的，神谷是一个有正义感的神偷，而寒羽良则成为一个见义勇为、武艺超群，而且足智多谋的职业侦探，这两个人有一个共同的特点，就是十分斗趣，总爱在女孩子面前开一些过火的玩笑。</p>
</body>
</html>
```

运行程序，效果如图 11-29 所示。

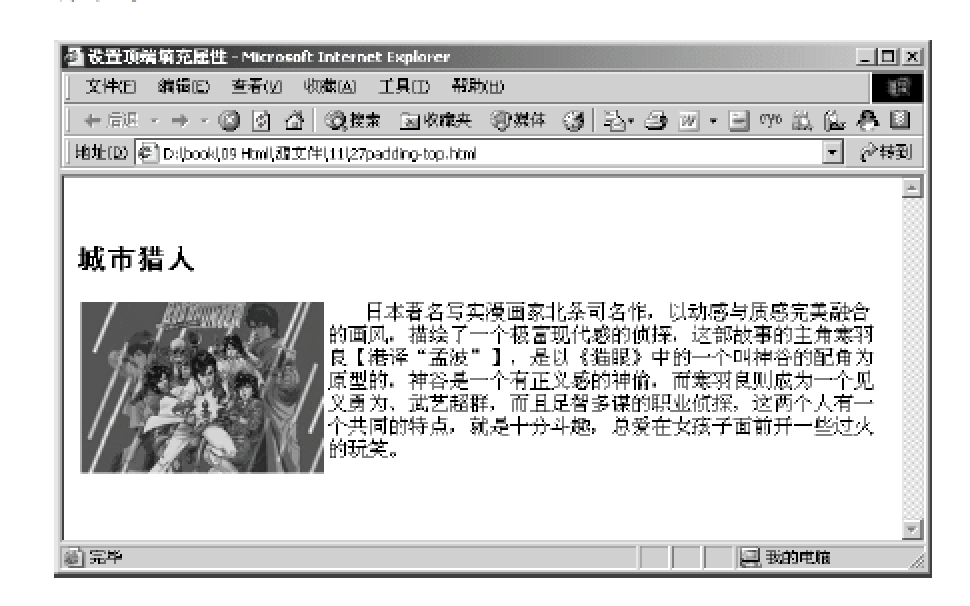


图 11-29　设置上补白

11.6.5　其他填充——padding-bottom、padding-right、padding-left

其他填充属性是指底端、左右两侧的补白值，其语法和使用方法同顶端填充类似，下面用一个实例说明其他填充的效果。

```
<html>
<head>
<title>设置其他填充属性</title>
<style>
<!--
      Body{padding-top:15pt; padding-bottom:13pt; padding-right:30pt; padding-left:20pt}
      h2 {font-family:黑体; font-size:16pt}
      p {font-size:11pt; text-indent:20pt}
-->
</style>
</head>
<body>
      <h2>城市猎人</h2>
      <p><img src="pic04.jpg" align="left">日本著名写实漫画家北条司名作，以动感与质感完美融合的画风，描绘了一个极富现代感的侦探，这部故事的主角寒羽良【港译“孟波”】，是以《猫眼》中的一个叫神谷的配角为原
```

```
型的，神谷是一个有正义感的神偷，而寒羽良则成为一个见义勇为、武艺超群，而且足智多谋的职业侦探，这两个人有一个共同的特点，就是十分斗趣，总爱在女孩子面前开一些过火的玩笑。</p>
</body>
</html>
```

运行程序，效果如图 11-30 所示。

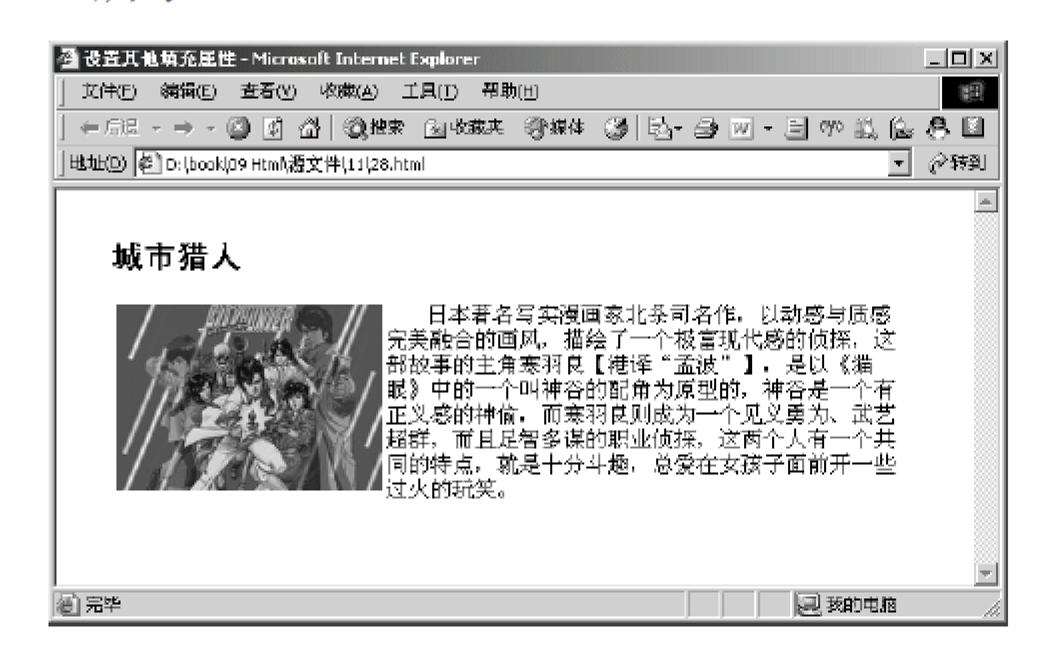


图 11-30　设置上、下、左、右补白

11.6.6　复合属性：填充——padding

与其他复合属性特别是边距属性 margin 的使用方式和用法类似，下面通过一个实例来说明填充属性的效果。

语法：margin: 长度值 | 百分比

实例代码：

```
<html>
<head>
<title>设置填充属性</title>
<style>
<!--
    Body{padding: 18pt 35pt 12pt}
    h2 {font-family:黑体; font-size:16pt}
    p {font-size:11pt; text-indent:20pt}
-->
</style>
</head>
<body>
    <h2>城市猎人</h2>
    <p><img src="pic04.jpg" align="left">日本著名写实漫画家北条司名作，以动感与质感完美融合的画风，描绘了一个极富现代感的侦探，这部故事的主角寒羽良【港译“孟波”】，是以《猫眼》中的一个叫神谷的配角为原型的，神谷是一个有正义感的神偷，而寒羽良则成为一个见义勇为、武艺超群，而且足智多谋的职业侦探，这两个人有一个共同的特点，就是十分斗趣，总爱在女孩子面前开一些过火的玩笑。</p>
</body>
</html>
```

运行代码，效果如图 11-31 所示。

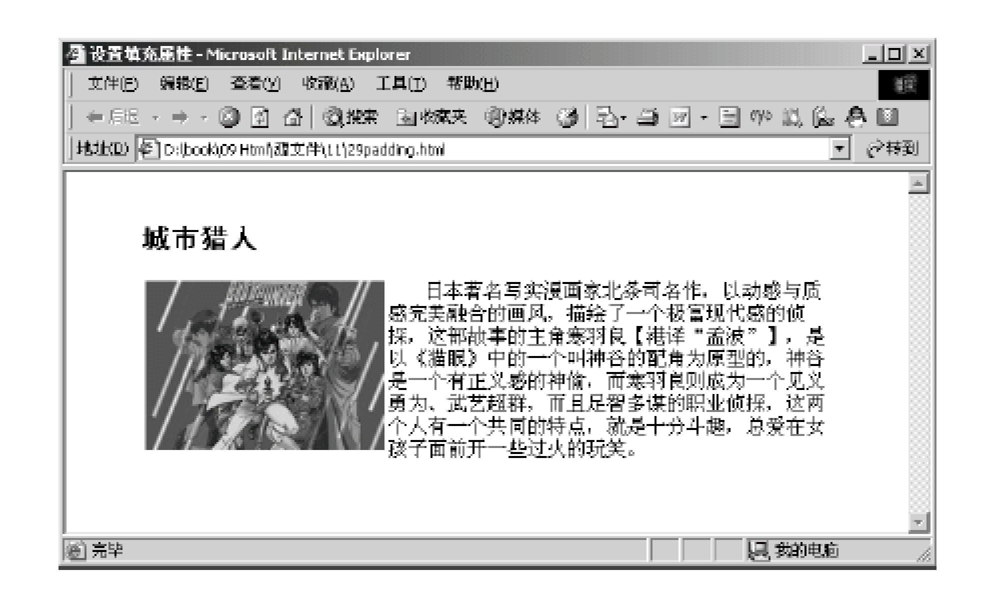


图 11-31　设置填充属性

11.7 边 框 属 性

边框属性控制元素所占用空间的边缘。例如，可将文本格式和定位属性应用到 <DIV> 元素，然后应用边框属性以创建元素周围的框，并将其设置为远离主要文本。在边框属性中，可以设置边框宽度、边框样式、边框颜色等。而每一类都包含 5 个属性，例如边框宽度其实具体可以分成上边框宽度 border-top-width、右边框宽度 border-right-width、下边框宽度 border-bottom-width、左边框宽度 border-left-width 以及宽度属性 border-width 等 5 个具体的属性。

边框属性中包含的具体属性见表 11-2。

表11-2　边框属性列表

属　　性	含　　义
border-top-width	设置上边框的宽度值
border-right-width	设置右边框的宽度值
border-bottom-width	设置下边框的宽度值
border-left-width	设置左边框的宽度值
border-width	复合属性，是设置边框宽度值的4个属性的略写
border-top-color	设置上边框的颜色
border-right-color	设置右边框的颜色
border-bottom-color	设置下边框的颜色
border-left-color	设置左边框的颜色
border-color	复合属性，是设置边框颜色的4个属性的略写
border-top-style	设置上边框的样式
border-right-style	设置右边框的样式
border-bottom-style	设置下边框的样式
border-left-style	设置左边框的样式
border-style	复合属性，是设置边框样式的4个属性的略写
boder-top	复合属性，可以同时设置上边框的宽度、颜色和样式
boder-right	复合属性，设置右边框的宽度、颜色和样式
boder-bottom	复合属性，设置下边框的宽度、颜色和样式
boder-left	复合属性，设置左边框的宽度、颜色和样式
boder	复合属性，相当于对上面所有属性的集合

由于边框属性的设置与边距、填充属性类似，为了便于理解和对比，每一种属性的上、下、左、右 4 个属性将会在一起讲解。

11.7.1　边框样式——border-style

边框样式属性用以定义边框的风格呈现式样，这个属性必须用于指定可见的边框。它可以对元素

分别设置上边框样式（border-top-style）、下边框样式（border-bottom-style）、左边框样式（border-left-style）和右边框样式（border-right-style）4 个属性，也可以使用复合属性边框样式（border-style）对边框样式的设置进行略写。

语法：border-style: 样式值
border-top-style: 样式值
border-right-style: 样式值
border-bottom-style: 样式值
border-left-style: 样式值

说明：样式可以取的值共有 9 种，见表 11-3。

表11-3　边框样式取值含义

取　　值	含　　义
none	不显示边框，为默认属性值
dotted	点线
dashed	虚线，也称为短线
solid	实线
double	双实线
groove	边框带有立体感的沟槽
ridge	边框成脊形
inset	使整个方框凹陷，即在边框内嵌入一个立体边框
outset	使整个方框凸起，即在边框外嵌入一个立体边框

注意： 虽然这几个属性的取值范围相同，但是上、下、左、右 4 个具体的边框样式属性都是设置一个值，而复合属性 border-style 可以设置 1 到 4 个值来设置元素的边框样式，而其不同个数的取值其含义与复合属性边距 margin、填充 padding 类似。

为了使读者在学习的过程中更加清晰地掌握这几种属性，下面在同一个实例中设置这 5 个属性的值来加以对比说明。

实例代码：

```
<html>
<head>
<title>设置边框样式</title>
<style>
<!--
    h2 {
        font-family:黑体;
        font-size:16pt;
        border-style: inset
      }
    p {
        font-size:11pt;
        text-indent:20pt;
        border-top-style: dotted;
```

```
            border-right-style: dashed;
            border-bottom-style: double;
            border-left-style: solid
        }
-->
</style>
</head>
<body>
    <h2>模拟人生 2：大学</h2>
    <p>模拟人生 2 却将游戏内容瞄准了丰富多彩的大学生活。在游戏中玩家可以完全体验到大学四年的校园生活，虽然美国和我们的校园生活和学习方式略有不同，不过玩家依然可以感受到考试带来的欢喜与悲伤；打工、参加社团带来的成就或失落；室友或导师帮你建立的自信和愤怒。一切梦想都从大学开始，无论是回忆还是期待。
    </p>
    <center><img src="pic05.jpg"></center>
</body>
</html>
```

运行程序，效果如图 11-32 所示。其中<h2>标记内的文字边框凹陷，而段落文字的 4 个方向的边框样式则各有不同。



图 11-32　设置边框样式

11.7.2　边框宽度——border-width

边框宽度用于设置元素边框的宽度值，其语法和用法都与边框样式的设置类似。

语法：border-width: 宽度值
　　border-top-width: 宽度值
　　border-right-width: 宽度值
　　border-bottom-width: 宽度值
　　border-left-width: 宽度值

取值范围：thin | medium | thick | <长度>

说明：在该语法中，这几种属性的取值范围相同。其中，medium 是默认宽度；thin 表示小于默认宽度，称为细边框；thick 大于默认宽度，称为粗边框；长度则是由数字和单位组成的长度值，不可为负值。

下面通过实例说明如何设置元素的边框宽度，实例代码如下：

```
<html>
```

```
<head>
<title>设置边框宽度</title>
<style>
<!--
      h2 {
            font-family:黑体;
            font-size:16pt;
            border-style: inset;
            border-width: 1px 5px 10px 3px
        }
      p {
            font-size:11pt;
            text-indent:20pt;
            /* 定义了段落文字的边框样式 */
            border-top-style: dotted;
            border-right-style: dashed;
            border-bottom-style: double;
            border-left-style: solid;
            /* 定义了段落文字的边框宽度*/
            border-top-width: 8px;
            border-right-width: 1px;
            border-bottom-width: 3px;
            border-left-width: 4px
        }
-->
</style>
</head>
<body>
      <h2>模拟人生 2：大学</h2>
      <p>模拟人生 2 却将游戏内容瞄准了丰富多彩的大学生活。在游戏中玩家可以完全体验到大学四年的校园生活，虽然美国和我们的校园生活和学习方式略有不同，不过玩家依然可以感受到考试带来的欢喜与悲伤；打工、参加社团带来的成就或失落；室友或导师帮你建立的自信和愤怒。一切梦想都从大学开始，无论是回忆还是期待。
       </p>
      <center><img src="pic05.jpg"></center>
</body>
</html>
```

运行这段代码，可以看到如图 11-33 所示的效果。

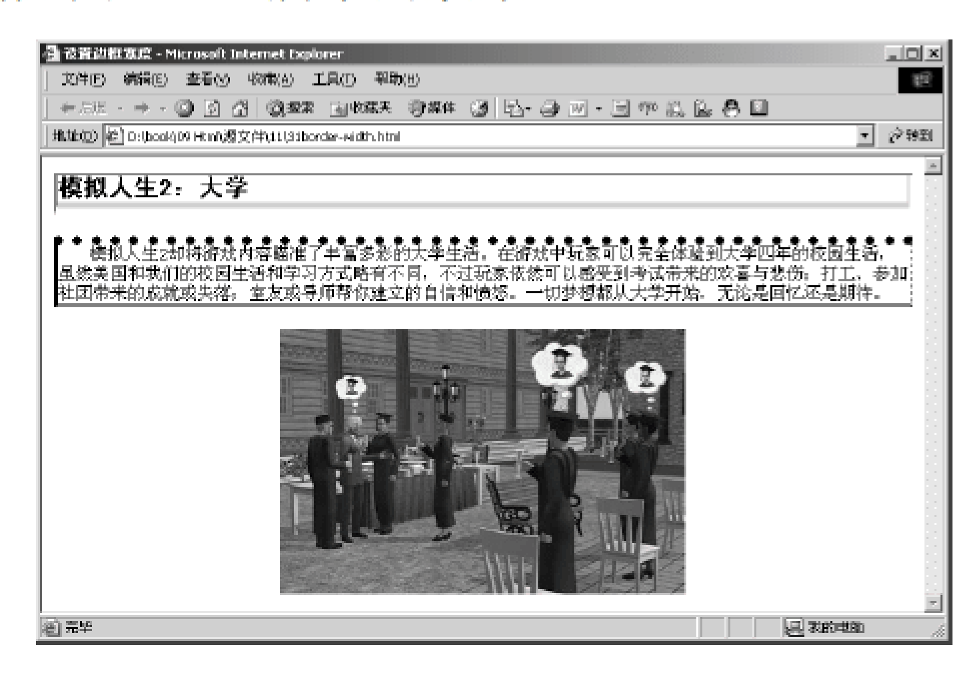


图 11-33　设置边框宽度

11.7.3　边框颜色——border-color

边框颜色属性用于定义边框的颜色，可以用 16 种颜色的关键字或 RGB 值来设置。可以对 4 个边框分别设置颜色，也可以使用复合属性 border-color 进行统一设置。对于使用边框颜色 border-color 属性，如果指定 1 种颜色，则表示 4 个边框是一种颜色；指定 2 种颜色，则定义顺序为上下、左右；指定 3 种颜色，顺序为上、左右、下；指定 4 种颜色，顺序则为上、右、下、左。

下面通过实例说明边框颜色的设置，实例代码如下：

```
<html>
<head>
<title>设置边框颜色</title>
<style>
<!--
    h2 {
        font-family:黑体;
        font-size:16pt;
        border-style: inset;
        border-width: 1px 5px 10px 3px;
        border-color: green red
    }
    p {
        font-size:11pt;
        text-indent:20pt;
        border-top-style: dotted;
        border-right-style: dashed;
        border-bottom-style: double;
        border-left-style: solid;
        border-top-width: 8px;
        border-right-width: 1px;
        border-bottom-width: 3px;
        border-left-width: 4px;
        /* 定义了段落文字的边框宽度*/
        border-top-color: #D00000;
        border-right-color: #00F;
        border-bottom-color: #FF00FF;
        border-left-color: maroon
    }
-->
</style>
</head>
<body>
    <h2>模拟人生 2：大学</h2>
    <p>模拟人生 2 却将游戏内容瞄准了丰富多彩的大学生活。在游戏中玩家可以完全体验到大学四年的校园生活，虽然美国和我们的校园生活和学习方式略有不同，不过玩家依然可以感受到考试带来的欢喜与悲伤；打工、参加社团带来的成就或失落；室友或导师帮你建立的自信和愤怒。一切梦想都从大学开始，无论是回忆还是期待。
    </p>
    <center><img src="pic05.jpg"></center>
</body>
</html>
```

运行这段代码，可以看到如图 11-34 所示的效果。

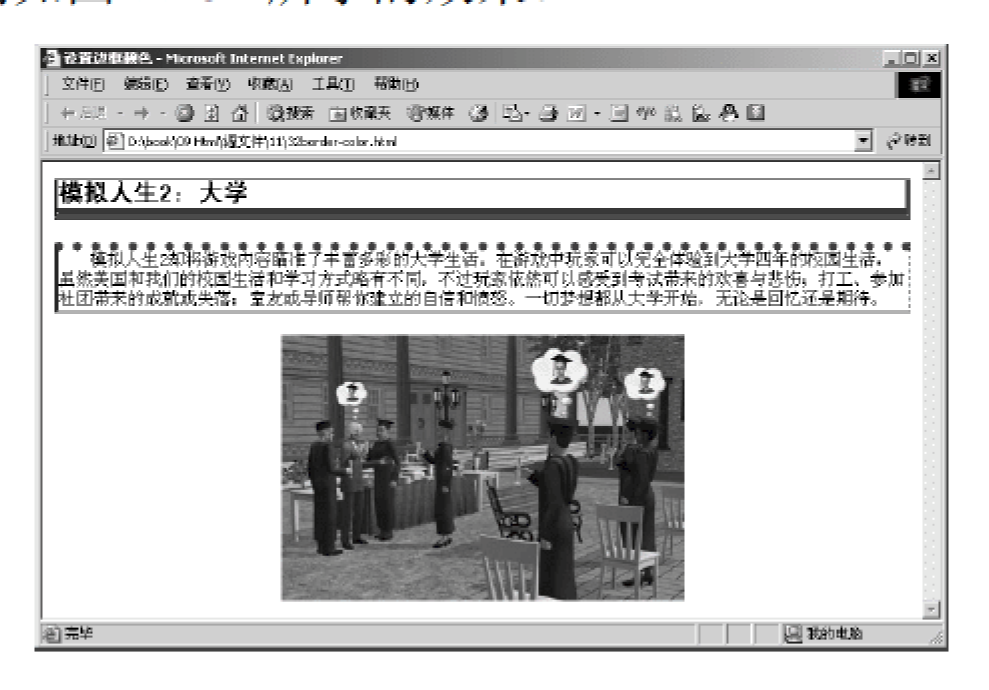


图 11-34　设置边框颜色

11.7.4　边框属性——border

边框属性用来设置一个元素的边框宽度、样式和颜色。边框属性只能设置 4 种边框，它所包含的 5 种属性（即上、下、左、右 4 个边框属性和一个总的边框属性）都是复合属性，在使用中只能给出一组边框宽度和样式。

语法：

border: <边框宽度> || <边框样式> || <颜色>
border-top: <上边框宽度> || <上边框样式> || <颜色>
border-right: <右边框宽度> || <右边框样式> || <颜色>
border-bottom: <下边框宽度> || <下边框样式> || <颜色>
border-left: <左边框宽度> || <左边框样式> || <颜色>

说明：在这些复合属性中，边框属性 border 只能同时设置 4 种边框，也只能给出一组边框的宽度和样式。而其他边框属性（如上边框属性 border-top）只能给出某一个边框的属性，包括宽度、样式和颜色。

实例代码：

```
<html>
<head>
<title>设置边框属性</title>
<style>
<!--
	h2 {
		font-family:黑体;
		font-size:16pt;
		border:5px dashed olive
	  }
	p {
		font-size:11pt;
		text-indent:20pt;
		/*  定义段落文字的边框属性*/
		border-top:3px dotted #00F;
```

```
            border-right:5px solid red;
            border-bottom:1px ridge #FF00FF;
            border-left:4px solid red
        }
-->
</style>
</head>
<body>
    <h2>模拟人生 2：大学</h2>
    <p>模拟人生 2 却将游戏内容瞄准了丰富多彩的大学生活。在游戏中玩家可以完全体验到大学四年的校园生活，虽然美国和我们的校园生活和学习方式略有不同，不过玩家依然可以感受到考试带来的欢喜与悲伤；打工、参加社团带来的成就或失落；室友或导师帮你建立的自信和愤怒。一切梦想都从大学开始，无论是回忆还是期待。
    </p>
    <center><img src="pic05.jpg"></center>
</body>
</html>
```

运行这段代码，可以看到如图 11-35 所示的效果。

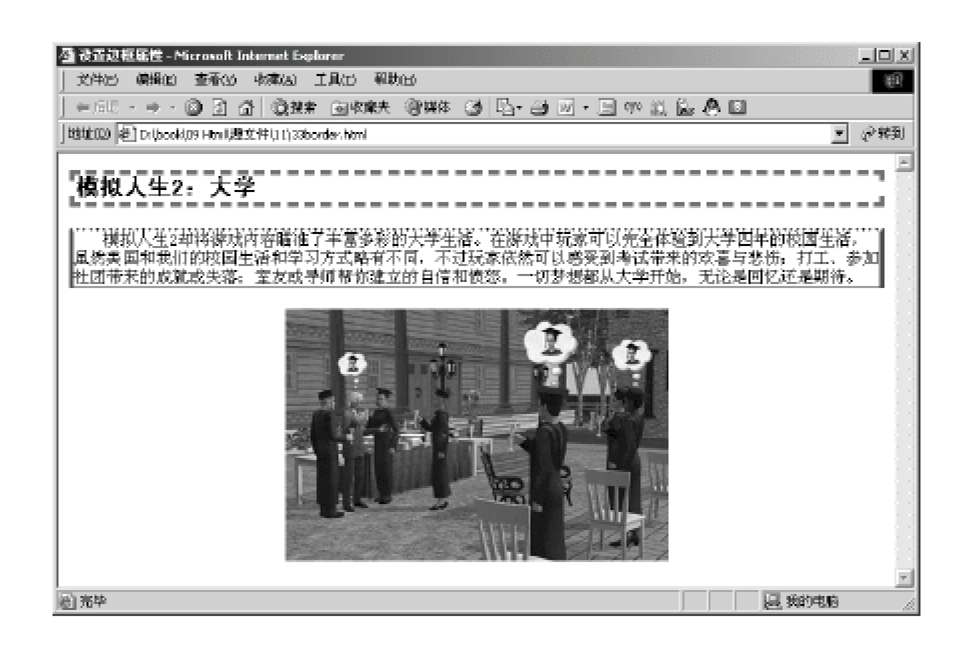


图 11-35　设置边框属性

11.8　定位及尺寸属性

定位属性控制网页所显示的整个元素的位置。例如，如果一个层元素（<div>标记）既包含文本又包含图片，则可用定位属性控制整个层元素的位置。可设置元素放置在页面中的绝对位置，也可设置为相对于其他元素的位置。定位属性主要通过相对定位和绝对定位两种方式定位。相对定位是指允许元素在相对于文档布局的原始位置上进行偏移，而绝对定位允许元素与原始的文档布局分离且任意定位。定位属性主要包括定位方式、层叠顺序等。

尺寸属性主要包括长度和宽度属性，用于确定元素的大小。下面对这些属性进行具体介绍。

11.8.1　定位方式——position

定位方式属性用于设定浏览器应如何来定位 HTML 元素。

语法：position : static | absolute | fixed | relative

说明：其中，static 表示无特殊定位，对象遵循 HTML 定位规则，是默认取值；absolute 表示采用绝对定位，需要同时使用 left、right、top、bottom 等属性进行绝对定位，而其层叠通过 z-index 属性定义，此时对象不具有边距，但仍有补白和边框；fixed 表示当页面滚动时，元素保持在浏览器视区内，其行为类似 absolute，在 IE5.5 中不支持该属性；relative 表示采用相对定位，对象不可层叠，但将依据 left、right、top、bottom 等属性设置在页面中的偏移位置。

注意： 当使用 absolute（绝对）定位元素时，该元素就被当作一个矩形覆盖物来格式化，格式化后的矩形区域就变成了一个可以放置其他 HTML 元素的容器，这个容器也就是层元素，它可以凌驾于 HTML 文档的布局之上，区域下面的文字和图形永远也无法环绕和透过该容器显示出来。

11.8.2　元素位置——top、right、bottom、left

元素位置属性与定位方式共同设置元素的具体位置。

语法：

top : auto | 长度值| 百分比

right : auto | 长度值| 百分比

bottom : auto | 长度值| 百分比

left : auto | 长度值 | 百分比

说明：这 4 个属性分别表示对象与其最近一个定位的父对象顶部、右侧、底部和左侧的相对位置，其中，auto 表示采用默认值，长度值则需要包含数字和单位，也可以使用百分数进行设置。

下面通过一个实例同时设置定位方式和元素位置属性，代码如下：

```
<html>
<head>
<title>对元素进行定位</title>
<style>
<!--
    h2 {
        font-family:黑体;
        font-size:16pt
     }
    .place {
        font-size:11pt;
        color:#CC0000;
        text-indent:20pt;
        position: absolute;
        top: 70pt;
        left: 30pt
     }
-->
</style>
</head>
<body>
    <h2>蜻蜓的眼泪</h2>
```

```
	<center><img src="pic06.jpg"></center>
	<div class=place>男孩跟女孩是一对热恋中的恋人，很不幸的是女孩在一次体操比赛中倒下了，医生确诊为：白血病。男孩每天都陪在女孩的病床前，看着女孩那苍白憔悴的脸，男孩哭了。哭的很伤心，男孩每天晚上都在女孩地病床前，为女孩折上几只千纸鹤，就这样过了几个月。女孩已经奄奄一息了，一直昏迷着，男孩还是重复地为女孩折千纸鹤，男孩折到 1000 只整的时候，病房被照亮了，天使忽然出现在男孩的面前，男孩被吓呆了，揉了揉哭肿了的眼睛，握着女孩的手，哭着对天使说，求你救救这个善良的女孩吧，天使对男孩说，只要你愿意，为女孩做 3 年蜻蜓，女孩就会慢慢地好起来，男孩毫不犹豫地答应了，天使消失了，同时男孩也变成了一只蜻蜓。后来……</div>
</body>
</html>
```

运行程序，可以看到<div>标记的定位效果如图 11-36 所示。

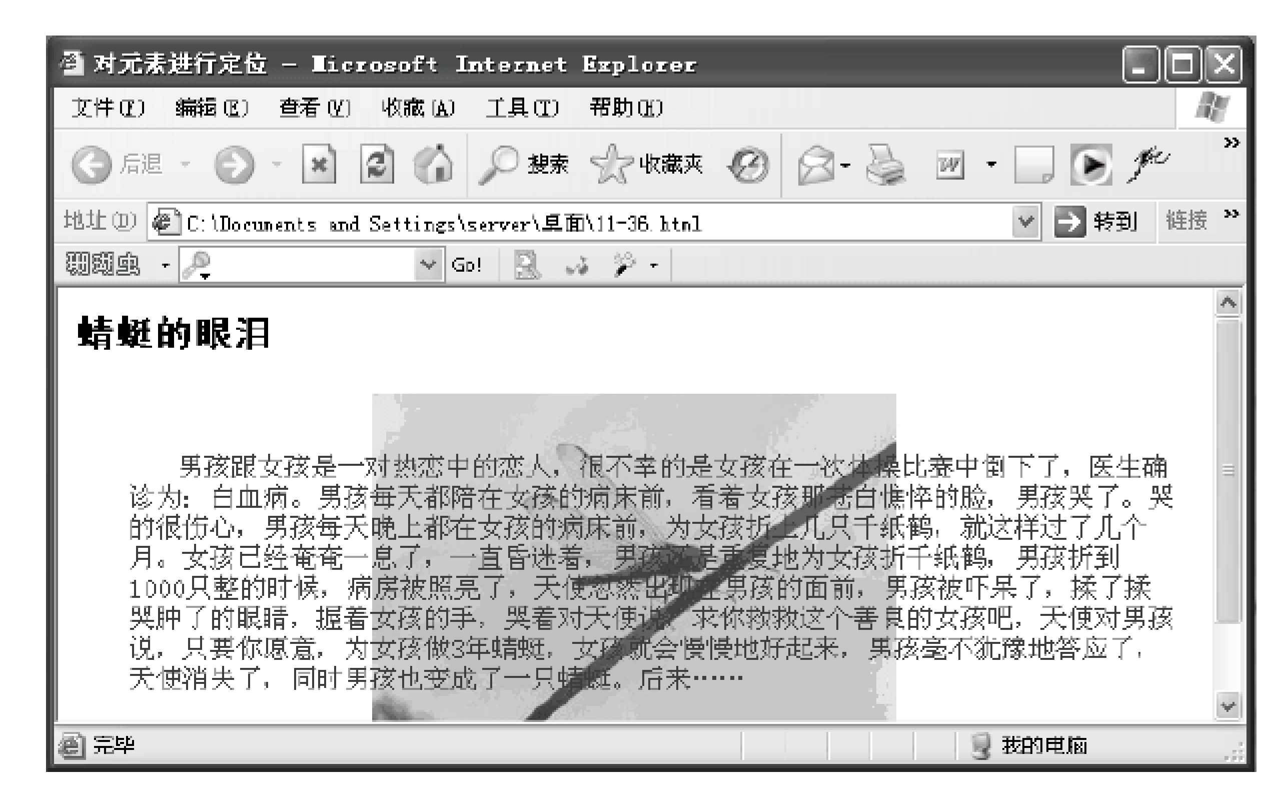


图 11-36　定位元素的效果

11.8.3　层叠顺序——z-index

层叠顺序属性用于设定层的先后顺序和覆盖关系，z-index 值高的层覆盖 z-index 值低的层。一般情况下，z-index 值为 1，表示该层位于最下层。

语法：z-index: auto | 数字

说明：当 z-index 取值为 auto 时，表示它遵循其父对象的定位属性；如果设置为数字，必须是无单位的整数值，可以取负值，但一般情况下都取正整数。

实例代码：

```
<html>
<head>
<title>设置层叠效果</title>
<style>
<!--
	h2 {
		font-family:黑体;
		font-size:16pt
	  }
	.lowdiv {
		position: absolute;
```

```
            top: 55pt;
            left: 230pt;
            z-index:1
        }
        .place {
            font-size:11pt;
            color:#CC0000;
            text-indent:20pt;
            position: absolute;
            top: 70pt;
            left: 30pt;
            z-index:2
        }
-->
</style>
</head>
<body>
    <h2>蜻蜓的眼泪</h2>
    <div class=lowdiv><img src="pic06.jpg"></div>
    <div class=place>
        男孩跟女孩是一对热恋中的恋人，很不幸的是女孩在一次体操比赛中倒下了，医生确诊为：白血病。男孩每天都陪在女孩的病床前，看着女孩那苍白憔悴的脸，男孩哭了。哭的很伤心，男孩每天晚上都在女孩地病床前，为女孩折上几只千纸鹤，就这样过了几个月。女孩已经奄奄一息了，一直昏迷着，男孩还是重复地为女孩折千纸鹤，男孩折到 1000 只整的时候，病房被照亮了，天使忽然出现在男孩的面前，男孩被吓呆了，揉了揉哭肿了的眼睛，握着女孩的手，哭着对天使说，求你救救这个善良的女孩吧，天使对男孩说，只要你愿意，为女孩做 3 年蜻蜓，女孩就会慢慢地好起来，男孩毫不犹豫地答应了，天使消失了，同时男孩也变成了一只蜻蜓。后来……
<br>
        <img src="pic06.jpg" width="110" align="right">
    </div>
</body>
</html>
```

运行代码，效果如图 11-37 所示。



图 11-37　两个层的层叠顺序

从实现的效果可以看出：应用了样式 lowdiv 类的层（简称 lowdiv 层）的 z-index 属性为 1，应用了样式 place 类的层（简称 place 层）的 z-index 属性为 2，lowdiv 层的内容（即图像）被 place 层的内容（包括文字和小的图像）所遮盖，这就证明了 z-index 属性值越高，它的层就越靠上。

11.8.4　浮动属性——float

浮动属性也称漂浮属性，用于将文字设置在某个元素的周围。它的功能相当于 img 元素的 align=left 和 align=right，但是 float 能应用于所有的元素，而不仅是图像和表格。

语法：float: left | right | none

说明：在该语法中，left 表示文字浮在元素左侧；right 则是浮在元素右侧；none 属于默认值，表示对象不浮动。

实例代码：

```
<html>
<head>
<title>设置浮动属性</title>
<style>
<!--
    h2 {
        font-family:黑体;
        font-size:16pt
     }
    .place {
        font-size:11pt;
        color:#CC0000;
     }
    img { float: right }
-->
</style>
</head>
<body>
    <h2>蜻蜓的眼泪</h2>
    <img src="pic06.jpg" width="130">
    <div class=place>
        男孩跟女孩是一对热恋中的恋人，很不幸的是女孩在一次体操比赛中倒下了，医生确诊为：白血病。男
孩每天都陪在女孩的病床前，看着女孩那苍白憔悴的脸，男孩哭了。哭的很伤心，男孩每天晚上都在女孩的病床
前，为女孩折上几只千纸鹤，就这样过了几个月。女孩已经奄奄一息了，一直昏迷着，男孩还是重复地为女孩折
千纸鹤，男孩折到 1000 只整的时候，病房被照亮了，天使忽然出现在男孩的面前，男孩被吓呆了，揉了揉哭肿了
的眼睛，握着女孩的手，哭着对天使说，求你救救这个善良的女孩吧，天使对男孩说，只要你愿意，为女孩做 3
年蜻蜓，女孩就会慢慢地好起来，男孩毫不犹豫地答应了，天使消失了，同时男孩也变成了一只蜻蜓。后来……
    </div>
</body>
</html>
```

运行代码，效果如图 11-38 所示。

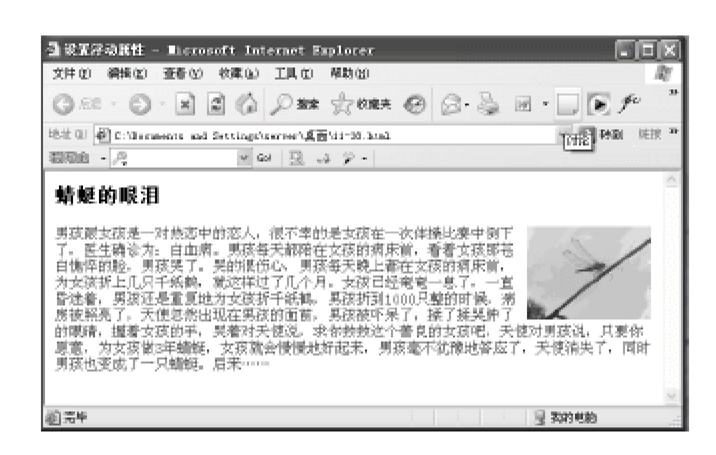


图 11-38　设置浮动属性

11.8.5　清除属性——clear

清除属性指定一个元素是否允许有其他元素漂浮在它的周围。

语法：clear: none | left | right | both

说明：none 表示允许两边都可以有浮动对象；left 表示不允许左边有浮动对象；right 表示不允许右边有浮动对象；both 则表示完全不允许有浮动对象。

实例代码：

```
<html>
<head>
<title>清除属性的效果</title>
<style>
<!--
	h2 {
		font-family:黑体;
		font-size:16pt
	  }
	.place {
		font-size:11pt;
		color:#CC0000;
		clear: right
	  }
	img { float: right }

-->
</style>
</head>
<body>
	<h2>蜻蜓的眼泪</h2>
	<img src="pic06.jpg" width="130">
	<div class=place>
		男孩跟女孩是一对热恋中的恋人，很不幸的是女孩在一次体操比赛中倒下了，医生确诊为：白血病。男孩每天都陪在女孩的病床前，看着女孩那苍白憔悴的脸，男孩哭了。哭的很伤心，男孩每天晚上都在女孩的病床前，为女孩折上几只千纸鹤，就这样过了几个月。女孩已经奄奄一息了，一直昏迷着，男孩还是重复地为女孩折千纸鹤，男孩折到 1000 只整的时候，病房被照亮了，天使忽然出现在男孩的面前，男孩被吓呆了，揉了揉哭肿了的眼睛，握着女孩的手，哭着对天使说，求你救救这个善良的女孩吧，天使对男孩说，只要你愿意，为女孩做 3 年蜻蜓，女孩就会慢慢地好起来，男孩毫不犹豫地答应了，天使消失了，同时男孩也变成了一只蜻蜓。后来……
	</div>
</body>
</html>
```

这一实例中本来设置图像漂浮在文字右侧，但由于文字设置了不允许右侧有漂浮对象，因此文字被移动到图像的下方，如图 11-39 所示。

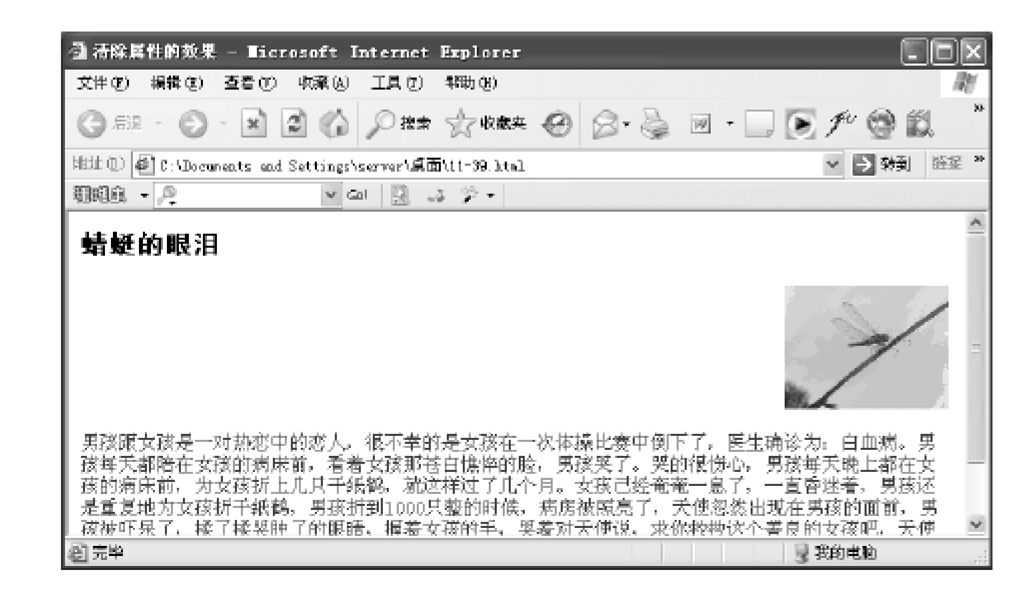


图 11-39　设置清除属性

11.8.6　可视区域——clip

可视区域用于设置层对象的可视区域，在区域外的部分是透明的。也可以认为通过设定了上下左右的边界值将对象裁切成一个矩形区域，在页面中只显示这个区域。但是只有在 position 的值设定为 absolute 时，该属性才能正常使用。

语法：clip : auto | rect（数值）

说明：auto 表示对象不裁切，rect (数值)中可以设定 4 个数字，它表示依据上、右、下、左的顺序，以对象左上角为(0,0)坐标计算的 4 个偏移数值，其中任何一个数值都可用 auto 替换，表示此边不裁切。

实例代码：

```
<html>
<head>
<title>设定可视区域属性</title>
<style>
<!--
	h2 {
		font-family:黑体;
		font-size:16pt
	  }
	.lowdiv {
		position: absolute;
		top: 55pt;
		left: 220pt;
		z-index:1
	  }
	.place {
		font-size:11pt;
		color:#CC0000;
		text-indent:20pt;
		position: absolute;
		top:70pt;
		left:15pt;
		z-index:2;
		clip: rect(auto 16cm 160px 3cm)
	  }
-->
</style>
</head>
<body>
	<h2>蜻蜓的眼泪</h2>
	<div class=lowdiv><img src="pic06.jpg"></div>
	<div class=place>
		男孩跟女孩是一对热恋中的恋人，很不幸的是女孩在一次体操比赛中倒下了，医生确诊为：白血病。男
孩每天都陪在女孩的病床前，看着女孩那苍白憔悴的脸，男孩哭了。哭的很伤心，男孩每天晚上都在女孩的病床
前，为女孩折上几只千纸鹤，就这样过了几个月。女孩已经奄奄一息了，一直昏迷着，男孩还是重复地为女孩折
千纸鹤，男孩折到 1000 只整的时候，病房被照亮了，天使忽然出现在男孩的面前，男孩被吓呆了，揉了揉哭肿了
```

```
的眼睛，握着女孩的手，哭着对天使说，求你救救这个善良的女孩吧，天使对男孩说，只要你愿意，为女孩做 3
年蜻蜓，女孩就会慢慢地好起来，男孩毫不犹豫地答应了，天使消失了，同时男孩也变成了一只蜻蜓。后来……
<br>
        <img src="pic06.jpg" width="110" align="right">
    </div>
</body>
</html>
```

运行代码，效果如图 11-40 所示，其中，同时包含文字和图像的层被裁切。其中设定左上角为原点 0.0，上侧不裁切，左侧从横坐标为 3cm 的位置裁切，右侧从左起横坐标为 16cm 的位置裁切，向下则从原点起 160px 的位置裁切。

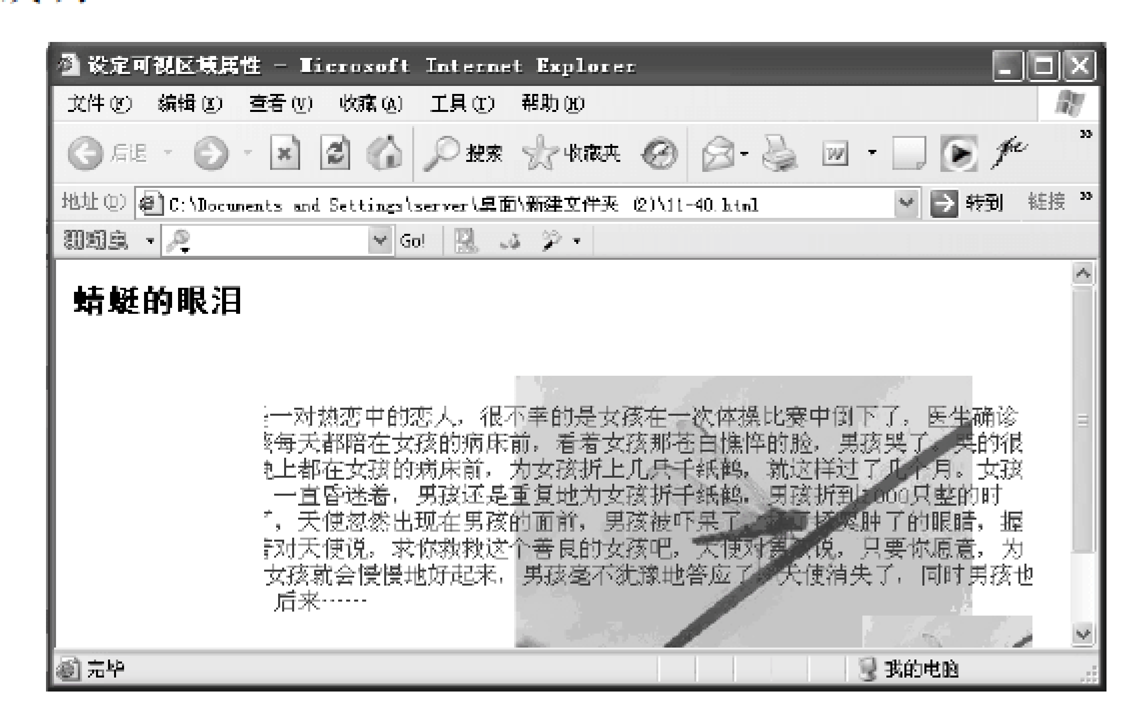


图 11-40　设定可视区域

11.8.7　设定大小——width、height

height 和 width 分别用于设定层的高度和宽度。

语法：

width : auto | 长度

height: auto | 长度

说明：此处，auto 表示自动设定长度，一般以层包含的内容为准，如果设定确切的长度，需要设定数值和单位。

注意： 对于一个层来说，一般只能设定宽度或高度中的一个值，另外一个值则根据内容自动确定。如果同时设定两个值，则需要同时定义后面将要讲解的 overflow 属性。

实例代码：

```
<html>
<head>
<title>设定层的宽度</title>
<style>
<!--
    h2 {
        font-family:黑体;
        font-size:16pt
```

```
        }
    .lowdiv {
            position: absolute;
            top: 55pt;
            left: 220pt;
            z-index:1
        }
    .place {
            font-size:11pt;
            color:#CC0000;
            text-indent:20pt;
            position: absolute;
            top:70pt;
            left:35pt;
            z-index:2;
            width: 310pt
        }
-->
</style>
</head>
<body>
    <h2>蜻蜓的眼泪</h2>
    <div class=lowdiv><img src="pic06.jpg"></div>
    <div class=place>
        男孩跟女孩是一对热恋中的恋人，很不幸的是女孩在一次体操比赛中倒下了，医生确诊为：白血病。男孩每天都陪在女孩的病床前，看着女孩那苍白憔悴的脸，男孩哭了。哭的很伤心，男孩每天晚上都在女孩的病床前，为女孩折上几只千纸鹤，就这样过了几个月。女孩已经奄奄一息了，一直昏迷着，男孩还是重复地为女孩折千纸鹤，男孩折到 1000 只整的时候，病房被照亮了，天使忽然出现在男孩的面前，男孩被吓呆了，揉了揉哭肿了的眼睛，握着女孩的手，哭着对天使说，求你救救这个善良的女孩吧，天使对男孩说，只要你愿意，为女孩做 3 年蜻蜓，女孩就会慢慢地好起来，男孩毫不犹豫地答应了，天使消失了，
同时男孩也变成了一只蜻蜓。后来……<br>
    </div>
</body>
</html>
```

运行程序，效果如图 11-41 所示，其中设定了文字层的宽度为 310pt。



图 11-41　设置层的大小

11.8.8　超出范围——overflow

超出范围属性用于设定当层的内容超出所能容纳的范围时的显示属性。

语法：overflow : visible | auto | hidden | scroll

说明：visible 表示不剪切内容也不添加滚动条；auto 是<body>对象和<textarea>对象的默认值，它在需要时剪切内容并添加滚动条；hidden 表示不显示超过对象尺寸的内容；scroll 则表示总显示滚动条。

注意： 如果显式声明 visible 为默认值，对象将被剪切为包含对象的窗口或框架的大小，并且 clip 属性设置将失效。

实例代码：

```
<html>
<head>
<title>设定超出范围属性</title>
<style>
<!--
	h2 {
		font-family:黑体;
		font-size:16pt
	  }
	.place {
		font-size:11pt;
		color:#CC0000;
		text-indent:20pt;
		position: absolute;
		top: 70pt;
		left: 30pt;
		height: 130px;
		width: 370px;
		overflow: hidden;
	  }
-->
</style>
</head>
<body>
	<h2>蜻蜓的眼泪</h2>
	<center><img src="pic06.jpg"></center>
	<div class=place>男孩跟女孩是一对热恋中的恋人，很不幸的是女孩在一次体操比赛中倒下了，医生确诊为：白血病。男孩每天都陪在女孩的病床前，看着女孩那苍白憔悴的脸，男孩哭了。哭的很伤心，男孩每天晚上都在女孩的病床前，为女孩折上几只千纸鹤，就这样过了几个月。女孩已经奄奄一息了，一直昏迷着，男孩还是重复地为女孩折千纸鹤，男孩折到 1000 只整的时候，病房被照亮了，天使忽然出现在男孩的面前，男孩被吓呆了，揉了揉哭肿了的眼睛，握着女孩的手，哭着对天使说，求你救救这个善良的女孩吧，天使对男孩说，只要你愿意，为女孩做 3 年蜻蜓，女孩就会慢慢地好起来，男孩毫不犹豫地答应了，天使消失了，同时男孩也变成了一只蜻蜓。后来……</div>
</body>
</html>
```

运行程序的效果如图 11-42 所示。



图 11-42　设置超出范围的显示属性

11.8.9　可见属性——visibility

可见属性用于设定嵌套层的显示属性，此属性可以将嵌套层隐藏，但仍然为隐藏对象保留其占据的物理空间。如果希望对象为可视，其父对象也必须是可视的。

语法：visibility : inherit | visible |hidden

说明：在该语法中，inherit 表示继承上一个父对象的可见性，即如果父对象可见，则该对象也可见，反之则不可见；visible 表示对象是可见的；hidden 表示对象隐藏。

实例代码：

```
<html>
<head>
<title>设置可见属性</title>
<style>
<!--
	h2 {
		font-family:黑体;
		font-size:16pt
	  }
	.lowdiv {
		position: absolute;
		top: 40pt;
		left: 240pt;
		z-index:1
	  }
	.place {
		font-size:11pt;
		color:#CC0000;
		text-indent:20pt;
		position: absolute;
		top: 50pt;
		left: 20pt;
		z-index:2;
		visibility : hidden
	  }
	img{ visibility : inherit }
-->
</style>
</head>
<body>
	<h2>蜻蜓的眼泪</h2>
	<div class=lowdiv><img src="pic06.jpg"></div>
	<div class=place>
		男孩跟女孩是一对热恋中的恋人，很不幸的是女孩在一次体操比赛中倒下了，医生确诊为：白血病。男孩每天都陪在女孩的病床前，看着女孩那苍白憔悴的脸，男孩哭了。哭的很伤心，男孩每天晚上都在女孩的病床前，为女孩折上几只千纸鹤，就这样过了几个月。女孩已经奄奄一息了，一直昏迷着，男孩还是重复地为女孩折千纸鹤，男孩折到 1000 只整的时候，病房被照亮了，天使忽然出现在男孩的面前，男孩被吓呆了，揉了揉哭肿了的眼睛，握着女孩的手，哭着对天使说，求你救救这个善良的女孩吧，天使对男孩说，只要你愿意，为女孩做 3 年蜻蜓，女孩就会慢慢地好起来，男孩毫不犹豫地答应了，天使消失了，同时男孩也变成了一只蜻蜓。后来……
<br>
		<img src="pic06.jpg" width="110" align="right">
	</div>
</body>
</html>
```

运行程序，可以看到应用了 place 类的层不可见，而图像 img 的 visibility 属性被设定为 inherit，因此 place 层中的图像也不可见，如图 11-43 所示。

如果将代码中的

```
img{ visibility : inherit }
```

更改为

```
img{ visibility : visible }
```

运行效果则如图 11-44 所示。由于设定了 img 的可见属性为 visible，因此虽然小图像所在的层被隐藏，但是图像仍然可见。



图 11-43　设定可见属性



图 11-44　调整可见属性

11.9　列 表 属 性

列表属性主要用于设置列表项的样式，包括符号、缩进等。

11.9.1　列表符号——list-style-type

列表符号属性用于设定列表项的符号。

语法：list-style-type: <值>

说明：可以设置多种符号作为列表项的符号，其具体取值范围见表 11-4。

表11-4　列表符号的取值

符号的取值	含　义
none	不显示任何项目符号或编号
disc	以实心圆形●作为项目符号
circle	以空心圆形○作为项目符号
square	以实心方块■作为项目符号
decimal	以普通阿拉伯数字1、2、3…作为项目编号

续表

符号的取值	含　义
lower-roman	以小写罗马数字i、ii、iii…作为项目编号
upper-roman	以大写罗马数字I、II、III…作为项目编号
lower-alpha	以小写英文字母a、b、c…作为项目编号
upper-alpha	以大写英文字母A、B、C…作为项目编号

实例代码：

```
<html>
<head>
<title>设定列表符号</title>
<style>
<!--
        body{ font-size: 12pt}
        h2 {
                font-family:黑体;
                font-size:16pt
           }
        ol{ list-style-type: square }
-->
</style>
</head>
<body>
        <h2>网页创作的相关知识：</h2>
        <ol >
                <li> HTML 是英文 Hyper Text Markup Language 的缩写，即超文本标志语言
                <li> CSS 是 Cascading Style Sheets（层叠样式表）的简称，是一种设计网页样式的工具
                <li> JavaScript 是一种新的描述语言，此语言可以被嵌入 HTML 的文件之中
        </ol>
</body>
</html>
```

运行程序，效果如图 11-45 所示。

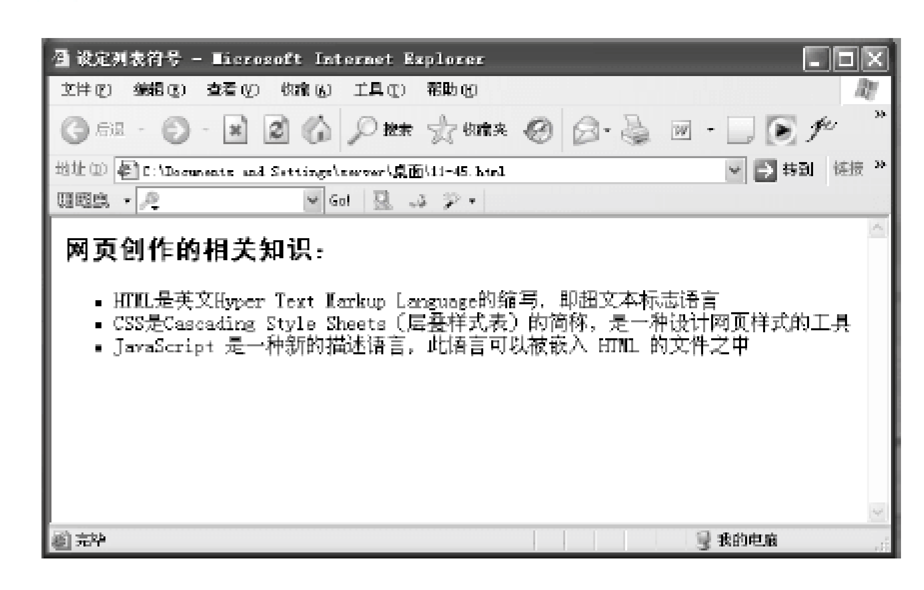


图 11-45　设定列表符号

11.9.2　图像符号——list-style-image

图像符号属性使用图像作为列表项目符号，以美化页面。

语法：list-style-img: list-style-image : none | url(图像地址)

说明：none 表示不指定图像；url 则使用绝对或相对地址指定作为符号的图像。

实例代码：

```
<html>
<head>
<title>设定图像符号</title>
<style>
<!--
    body{ font-size: 12pt}
    h2 {
        font-family:黑体;
        font-size:16pt
      }
    ol{
        list-style-type: square;
        list-style-image:url(pic07.jpg)
      }
-->
</style>
</head>
<body>
    <h2>网页创作的相关知识：</h2>
    <ol >
        <li> HTML 是英文 Hyper Text Markup Language 的缩写，即超文本标志语言
        <li> CSS 是 Cascading Style Sheets（层叠样式表）的简称，是一种设计网页样式的工具
        <li> JavaScript  是一种新的描述语言，此语言可以被嵌入  HTML  的文件之中
    </ol>
</body>
</html>
```

运行程序，效果如图 11-46 所示。

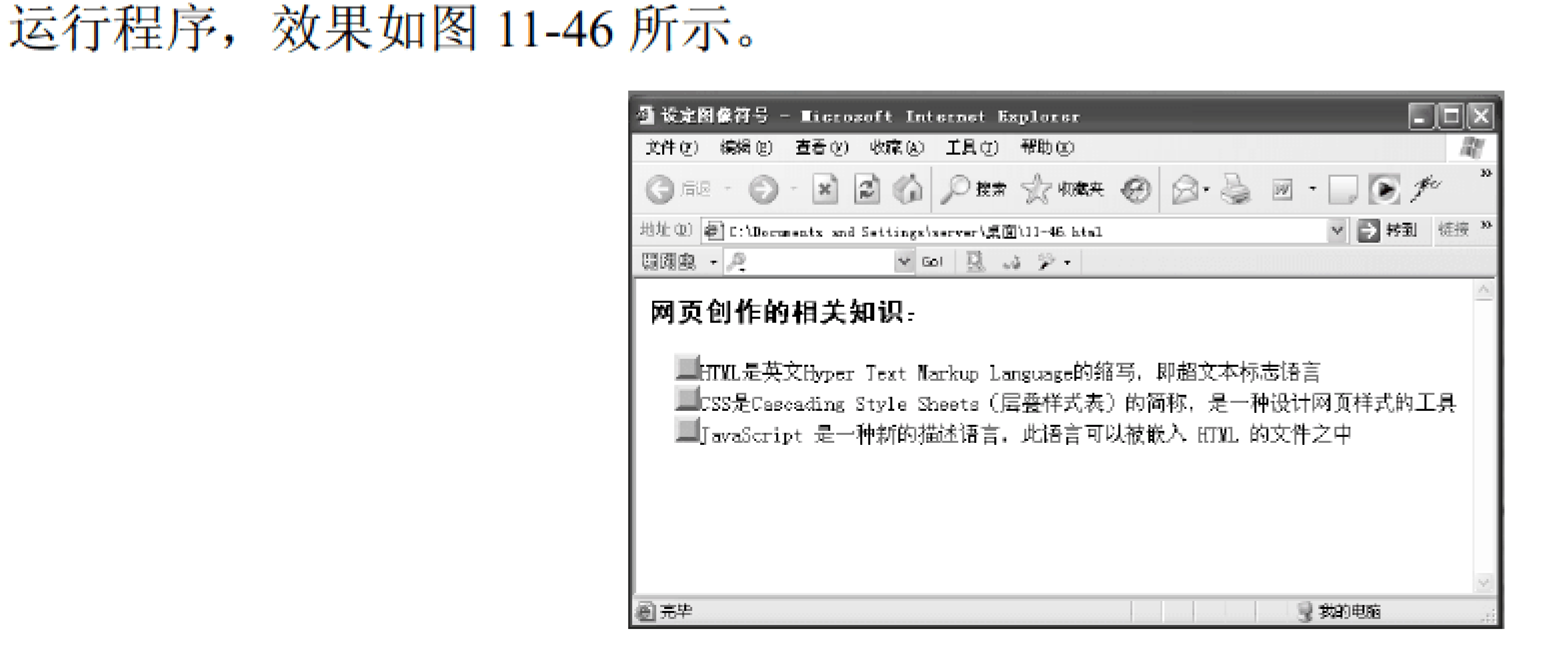


图 11-46　设置图像符号

11.9.3　列表缩进——list-style-position

列表缩进属性用于设定列表缩进的设置。

语法：list-style-position: outside | inside

说明：outside 表示列表项目标记放置在文本以外，且环绕文本不根据标记对齐；inside 是列表的默认属性，表示列表项目标记放置在文本以内，且环绕文本根据标记对齐。

实例代码：

```
<html>
<head>
<title>设定列表缩进</title>
<style>
<!--
    body{ font-size: 12pt}
    h2 {
        font-family:黑体;
        font-size:16pt
      }
    ol{
        list-style-type: square;
        list-style-image:url(pic07.jpg);
        list-style-position: inside
      }
-->
</style>
</head>
<body>
    <h2>网页创作的相关知识：</h2>
    <ol >
        <li> HTML 是英文 Hyper Text Markup Language 的缩写，即超文本标志语言
        <li> CSS 是“Cascading Style Sheet”的缩写，可以翻译为“层叠样式表”或“级联样式表”，即“样式表”，是一种设计网页样式的工具
        <li> JavaScript 是一种新的描述语言，此语言可以被嵌入 HTML 的文件之中
    </ol>
</body>
</html>
```

运行代码，可以看到设置缩进为 inside 的效果如图 11-47 所示。

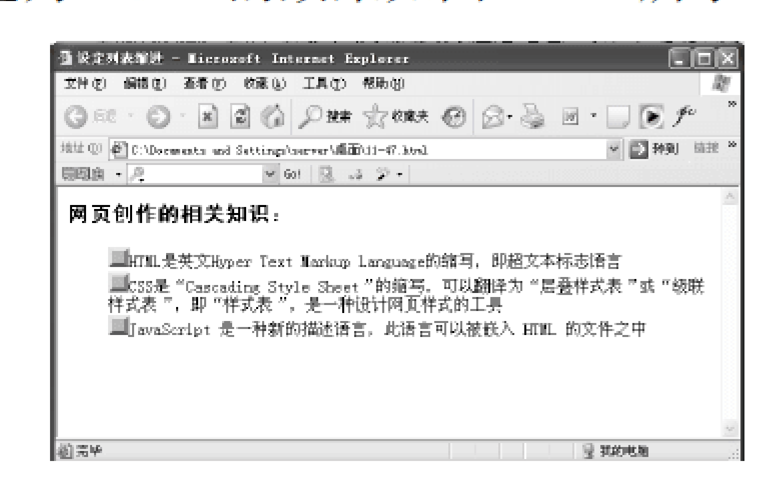


图 11-47　设置列表缩进

11.9.4　复合属性：列表——list-style

列表函数 list-style 是以上 3 种列表属性的组合。

将上一个实例使用 list-style 实现，代码如下：

```
<html>
<head>
<title>设定列表缩进</title>
<style>
<!--
      body{ font-size: 12pt}
      h2 {
            font-family:黑体;
            font-size:16pt
         }
      ol{ list-style: square inside url(pic07.jpg); }
-->
</style>
</head>
<body>
      <h2>网页创作的相关知识：</h2>
      <ol >
            <li> HTML 是英文 Hyper Text Markup Language 的缩写，即超文本标志语言
            <li> CSS 是“Cascading Style Sheet”的缩写，可以翻译为“层叠样式表”或“级联样式表”，即“样式表”，是一种设计网页样式的工具
            <li> JavaScript 是一种新的描述语言，此语言可以被嵌入 HTML 的文件之中
      </ol>
</body>
</html>
```

运行程序，会发现本实例能实现和上一个实例相同的效果，而且更加简便。

11.10　光标属性——cursor

光标属性是 CSS 中用于专门设置在对象上移动的鼠标指针所采用的光标形状。

语法：cursor : auto | 形状取值 | url(图像地址)

说明：在该语法中包含 3 种类型的取值，auto 表示根据页面的内容自动选择光标形状；url(图像)则表示采用自定义的图像作为光标形状；形状取值则是系统预定义的几种光标形状，其具体含义见表 11-5。

表11-5　光标形状取值

光标形状取值	具 体 含 义
hand	手形
crosshair	交叉十字形
text	文本选择形状
default	默认的箭头形状
help	带有问号的箭头

续表

光标形状取值	具 体 含 义
e-resize	向东的箭头↔
ne-resize	向东北的箭头⤢
n-resize	向北的箭头↕
nw-resize	向西北的箭头⤡
w-resize	向西的箭头↔
sw-resize	向西南的箭头⤢
s-resize	向南的箭头↕
se-resize	向东南的箭头⤡

实例代码：

```
<html>
<head>
<title>设定光标形状</title>
<style>
<!--
    h2 {
        font-family:黑体;
        font-size:16pt;
        cursor:text
      }
    img {cursor:hand}
-->
</style>
</head>
<body>
    <h2>一幅漂亮的图像</h2>
    <img src="pic06.jpg">
</body>
</html>
```

运行程序，看到当鼠标位于<h2>标记的内容上时，光标变为文本选择形状，如图 11-48 所示；而当鼠标位于图片上时，光标变为手形，如图 11-49 所示。



图 11-48　文本光标效果



图 11-49　手形光标效果

11.11 滤镜属性

filter 是微软对 CSS 的扩展，与 Photoshop 中的滤镜相似，它可以用很简单的方法对页面中的文字进行特效处理。使用滤镜属性可以把一些特别的效果添加到 HTML 元素中，使页面更加美观。但一般情况下，滤镜属性需要应用在已知大小的块级元素中，才能达到很好的效果。滤镜的基本语法如下：

filter：滤镜名称（参数 1，参数 2，…）

下面介绍几种常见的滤镜效果的具体设置方法。

11.11.1 不透明度——alpha

alpha 滤镜用于设置图片或文字的不透明度。它是把一个目标元素与背景混合，通俗地说就是一个元素的透明度。通过指定坐标，可以指定点、线、面的透明度。

语法：filter: alpha(参数 1=参数值, 参数 2=参数值,…)

说明：alpha 属性包括很多参数，见表 11-6。

表11-6　alpha属性的参数设置

参　数	具体含义及取值
opacity	代表透明度水准，默认的范围是从0～100，表示透明度的百分比。也就是说，0代表完全透明，100代表完全不透明
finishopacity	是一个可选参数，如果要设置渐变的透明效果，可以使用该参数来指定结束时的透明度。范围也是0～100
style	参数指定了透明区域的形状特征。其中0代表统一形状、1代表线形、2代表放射状、3代表长方形
startx	代表渐变透明效果的开始X坐标
starty	代表渐变透明效果的开始Y坐标
finishx	代表渐变透明效果的结束X坐标
finishy	代表渐变透明效果的结束Y坐标

实例代码：

```
<html>
<head>
<title>设置图像的透明效果</title>
<style>
<!--
      h2{ font-family: "黑体"; font-size:15pt}
      body{font-size:12pt}
      .alphaall{filter:alpha(opacity=50)}
      .alpharad{
      filter:alpha(opacity=0,finishopacity=100,style=2,startx=0,starty=5,finishx=200,finishy=185)
         }
-->
```

```
</style>
</head>
<body>
    <h2>下面是对同样的图像使用了不同 alpha 滤镜的效果</h2>
    <table border=1 bordercolor="#660000" cellpadding=5 width=600 align="center">
        <tr height=30>
            <td>原图</td>
            <td>整体设置 50%透明度的效果</td>
            <td>设置一个放射性的渐变透明效果</td>
        </tr>
        <tr height=240>
            <td><img src="pic08.jpg"></td>
            <td><img class=alphaall src="pic08.jpg"></td>
            <td><img class=alpharad src="pic08.jpg"></td>
        </tr>
    </table>
</body>
</html>
```

运行效果如图 11-50 所示。



图 11-50　设置图像不透明度的效果

11.11.2　动感模糊——blur

动感模糊滤镜用于在指定块级元素的方向和位置上产生动感模糊的效果。

语法：filter: blur(add=参数值, direction=参数值, strength=参数值)

说明：在语法中，add 参数是一个布尔判断，可取值为 true 或 false，它指定图片是否被改变成印象派的模糊效果；direction 参数用来设置模糊的方向，是按顺时针的方向以 45° 为单位进行累积的，例如 0 代表垂直向上，135 表示垂直向上开始顺时针旋转 135° 的方向，默认值是 270，即向左的方向；strength 值只能使用整数来指定，代表有多少像素的宽度将受到模糊影响，默认是 5 个。

注意： 动感模糊对 HTML 的块级元素（如层、图像等）有效，如果要对文字进行动感模糊，要首先将文字放置于一个块状元素内，如层，然后要确定这个块级元素的大小。

实例代码：

```
<html>
<head>
<title>设置元素的动感模糊</title>
<style type="text/css">
<!--
    body{ font-size:12pt}
    div.example {
        font-family:"黑体";
        font-size:16pt;
        width:500px;
        height:30px;
        filter:blur(add=ture,direction=135,strength=10);
```

```
        }
    .blurbig{filter:blur(strength=80);}
    .blursmall{filter:blur(direction=305,strength=20);}
-->
</style>
</head>
<body>
    <div class="example">下面是对同样的图像使用了不同 blur 滤镜的效果</div>
    <table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
        <tr height=70 align="center">
            <td width=220>原图</td>
            <td width=220>设置 strength 为 80 的动感模糊效果</td>
            <td width=220>设置方向为从垂直向上顺时针旋转 305 度，strength 设置为 20 的模糊效果</td>
        </tr>
        <tr height=260 align="center">
            <td><img src="pic09.jpg"></td>
            <td><img class="blurbig" src="pic09.jpg"></td>
            <td><img class="blursmall" src="pic09.jpg"></td>
        </tr>
    </table>
</body>
</html>
```

运行程序，效果如图 11-51 所示。



图 11-51　设置动感模糊的效果

11.11.3　对颜色进行透明处理——chroma

chroma 滤镜可以实现对所选择的颜色进行透明处理的效果。

语法：filter: chroma(color=颜色代码或颜色关键字)

说明：参数 color 即为要透明的颜色。

实例代码：

```
<html>
<head>
<title>设置颜色的透明处理</title>
```

```
<style type="text/css">
<!--
      body{ font-size:12pt}
      div.example {
            font-family:"黑体";
            font-size:16pt;
            width:500px;
            height:30px;
            filter:blur(add=ture,direction=135,strength=10);
        }
      .transone{ filter:chroma(color=#526373); }
      .transtwo{ filter:chroma(color=#52A5DE); }
-->
</style>
</head>
<body bgcolor="#FFDDBB">
      <div class="example">下面是对同样的图像使用了不同 blur 滤镜的效果</div>
      <table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
            <tr height=70 align="center">
                  <td width=220>原图</td>
                  <td width=220>女士的衣服颜色设为透明，露出背景色</td>
                  <td width=220>显示屏颜色为透明</td>
            </tr>
            <tr height=260 align="center">
                  <td><img src="pic10.gif"></td>
                  <td><img class=transone src="pic10.gif"></td>
                  <td><img class=transtwo src="pic10.gif"></td>
            </tr>
      </table>
</body>
</html>
```

运行程序，效果如图 11-52 所示。

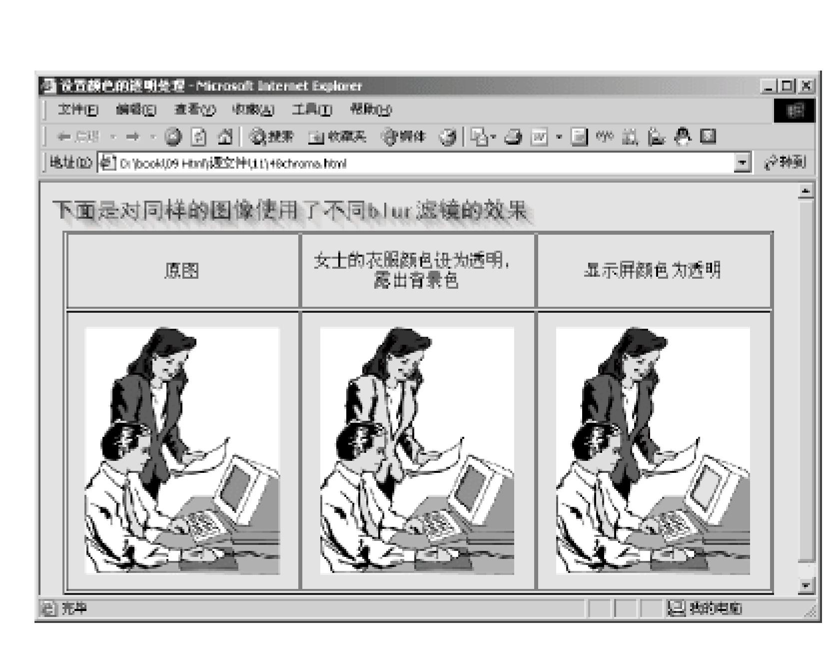


图 11-52　透明处理

11.11.4　设置阴影——dropShadow

dropShadow 滤镜在指定的方向和位置上产生阴影。

语法：filter: dropShadow(color=阴影颜色, offX=参数值, offY=参数值, positive=参数值)

说明：color 属性用于设置阴影的颜色；offX、offY 分别用于设置阴影相对图像移动的水平距离和垂直距离；positive 是一个布尔值（0 或 1），其中 0 指为透明像素生成阴影，1 指为不透明像素生成阴影。

实例代码：

```
<html>
<head>
<title>设置元素的阴影</title>
<style type="text/css">
```

```
<!--
        body{ font-size:12pt}
        div.example {
                font-family:"黑体";
                font-size:16pt;
                width:500px;
                height:30px;
                filter:dropshadow(color=blue, offx=5, offy=4, positive=1);
            }
                .dropshadow {filter:dropshadow(color="#212022",offx=15,offy=10, positive=1);}
-->
</style>
</head>
<body>
        <div class="example">下面是对同样的图像使用了阴影效果</div>
        <table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
                <tr height=30 align="center">
                        <td width=330>原图</td>
                        <td width=330>设置阴影效果</td>
                </tr>
                <tr height=260 align="center">
                        <td><img src="pic09.jpg"></td>
                        <td class=dropshadow><img src="pic09.jpg"></td>
                </tr>
        </table>
</body>
</html>
```

运行程序，效果如图 11-53 所示。



图 11-53　设置阴影的效果

11.11.5　对象的翻转——flipH、flipV

flipH 滤镜可以沿水平方向翻转对象，flipV 滤镜可以沿垂直方向翻转对象。

语法：

```
filter: flipH
```

filter: flipV

实例代码：

```
<html>
<head>
<title>翻转图像</title>
<style type="text/css">
<!--
    body{ font-size:12pt}
    h2{ font-family: "黑体"; font-size:15pt}
    .turnh{ filter: flipH}
    .turnv{ filter: flipV}
-->
</style>
</head>
<body>
    <h2>下面是对同样的图像进行了翻转</h2>
    <table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
        <tr height=30 align="center">
            <td width=220>原图</td>
            <td width=220>水平翻转效果</td>
            <td width=220>垂直翻转效果</td>
        </tr>
        <tr height=260 align="center">
            <td><img src="pic09.jpg"></td>
            <td class=turnh><img src="pic09.jpg"></td>
            <td class=turnv><img src="pic09.jpg"></td>
        </tr>
    </table>
</body>
</html>
```

运行程序，效果如图 11-54 所示。



图 11-54　对象的翻转

11.11.6　发光效果——glow

glow 滤镜可以实现在对象周围发光的效果。当对一个对象使用 glow 属性后，这个对象的边缘就会产生类似发光的效果。

语法：filter: glow(color=颜色代码, strength=强度值)

说明：color 参数用来指定发光的颜色；strength 参数用于设置强度，可以使用从 1～255 之间的任何整数来指定这个力度。

注意： 在使用 glow 滤镜的过程中，最好将要发光的元素放置到块级元素中使用，如层。

实例代码：

```
<html>
<head>
<title>发光效果</title>
<style type="text/css">
<!--
      body{ font-size:12pt}
      h2{ font-family: "黑体"; font-size:15pt; filter: glow(color=red, strength=50)}
      .glow{ filter: glow(color=blue, strength=8)}
      .glow2{ filter: glow(color=blue, strength=30)}
      div{width:210}
-->
</style>
</head>
<body>
      <h2>下面是对同样的图像使用了 glow 滤镜</h2>
      <table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
            <tr height=30 align="center">
                  <td width=200>原图</td>
                  <td width=220>strength 为 8 的蓝光效果</td>
                  <td width=240>strength 为 20 的蓝光效果</td>
            </tr>
            <tr height=260 align="center">
                  <td><img src="pic09.jpg"></td>
                  <td><divclass=glow><img
src="pic09.jpg"></div></td>
                  <td><divclass=glow2><img
src="pic09.jpg"></div></td>
            </tr>
      </table>
</body>
</html>
```

运行程序，效果如图 11-55 所示。



图 11-55　发光效果

11.11.7　灰度处理——gray

gray 滤镜是把一张图片变成灰度图。灰度也不需要设定参数，它去除目标的所有色彩，将其以灰度级别显示。

语法：filter:gray

实例代码：

```
<html>
<head>
<title>灰度处理</title>
<style type="text/css">
<!--
    body{ font-size:12pt}
    h2{ font-family: "黑体"; font-size:15pt; }
    .gray{ filter: gray}
    div{width:300}
-->
</style>
</head>
<body>
    <h2>下面是对同样的图像使用了灰度滤镜</h2>
    <table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
        <tr height=30 align="center">
            <td width=330>原图</td>
            <td width=330>灰度效果</td>
        </tr>
        <tr height=260 align="center">
            <td><img src="pic08.jpg"></td>
            <td><div class=gray><img src="pic08.jpg"></div></td>
        </tr>
    </table>
</body>
</html>
```

运行程序，效果如图 11-56 所示。



图 11-56　灰度处理的效果

11.11.8　反相——invert

invert 滤镜是把对象反相，也就是将其可视化属性全部反转，包括色彩、饱和度和亮度值。

语法：filter: invert

实例代码：

```
<html>
<head>
<title>反相处理</title>
<style type="text/css">
<!--
	body{ font-size:12pt}
	h2{ font-family: "黑体"; font-size:15pt; }
	.invert{ filter: invert}
	div{width:300}
-->
</style>
</head>
<body>
	<h2>下面是对同样的图像使用了反相效果</h2>
	<table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
		<tr height=30 align="center">
			<td width=330>原图</td>
			<td width=330>反相效果</td>
		</tr>
		<tr height=260 align="center">
			<td><img src="pic08.jpg"></td>
			<td><div class=invert><img src="pic08.jpg"></div></td>
		</tr>
	</table>
</body>
</html>
```

运行程序，效果如图 11-57 所示。



图 11-57　反相处理的效果

11.11.9　X 光片效果——xray

xray 滤镜用于加亮对象的轮廓，呈现所谓的“X”光片。X 光效果滤镜不需要设定参数，是一种很少见的滤镜。它可以像灰色滤镜一样去除图像的所有颜色信息，然后将其反转（黑白像素除外）。

语法：filter: xray

实例代码：

```
<html>
<head>
<title>X 光片效果</title>
<style type="text/css">
<!--
	body{ font-size:12pt}
	h2{ font-family: "黑体"; font-size:15pt; }
	.xray{ filter: xray}
	div{width:300}
-->
</style>
</head>
<body>
	<h2>下面是对同样的图像使用了 xray 滤镜</h2>
	<table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
		<tr height=30 align="center">
			<td width=330>原图</td>
			<td width=330>X 光片效果</td>
		</tr>
		<tr height=260 align="center">
			<td><img src="pic08.jpg"></td>
			<td><div class=xray><img src="pic08.jpg"></div></td>
		</tr>
	</table>
</body>
</html>
```

运行程序，效果如图 11-58 所示。

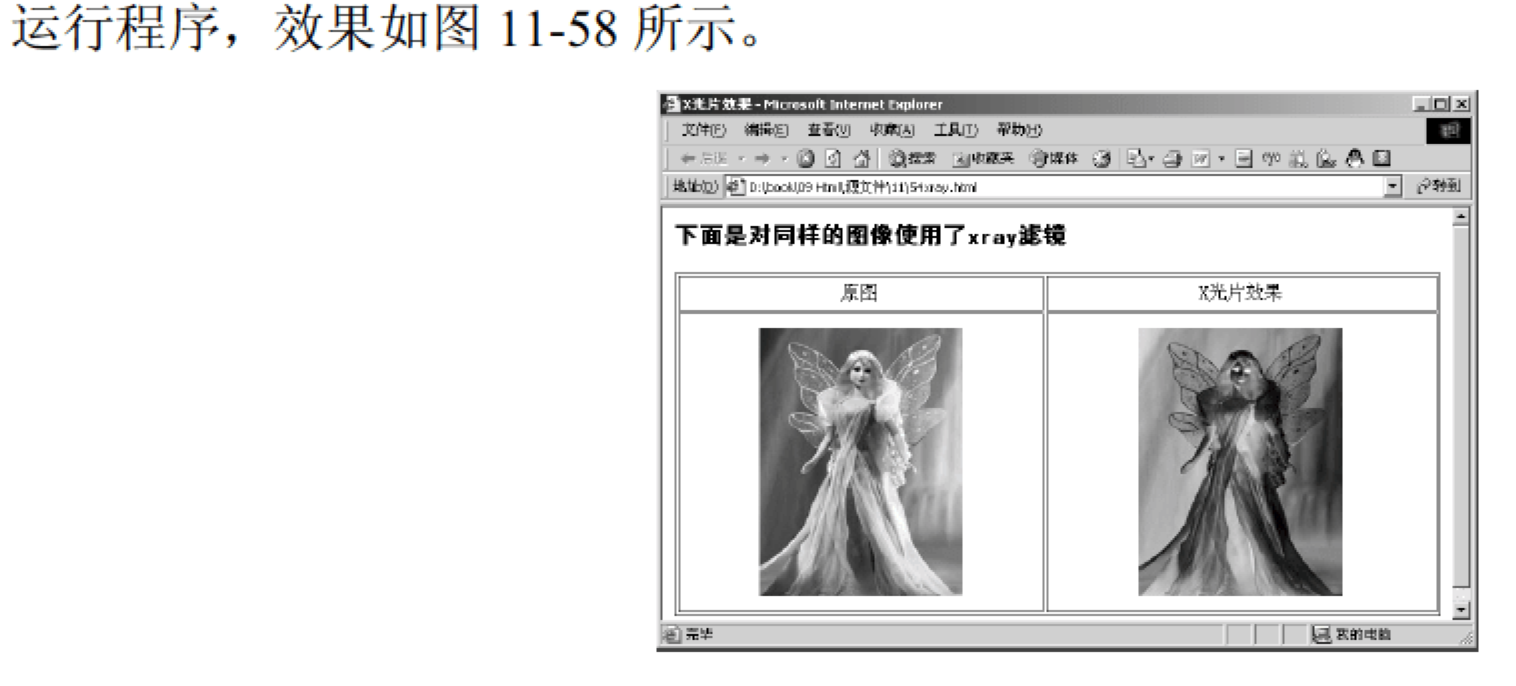


图 11-58　X 光片的效果

11.11.10　遮罩效果——mask

使用 mask 滤镜可以为对象建立一个覆盖于表面的膜。该滤镜将可看见的像素遮蔽，将看不见的像素以指定的颜色显示。

语法：filter:mask(color=颜色代码)

说明：这里的颜色是最后遮罩显示的颜色，而这种滤镜对 GIF 图像支持得最好。

实例代码：

```
<html>
<head>
<title>遮罩效果</title>
<style type="text/css">
<!--
	body{ font-size:12pt}
	h2{ font-family: "黑体"; font-size:15pt; }
	.mask{filter:mask(color=#FFAA00)}
-->
</style>
</head>
<body>
	<h2>下面是对同样的图像使用了遮罩滤镜</h2>
	<table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
		<tr height=30 align="center">
			<td width=330>原图</td>
			<td width=330>遮罩效果</td>
		</tr>
		<tr height=260 align="center">
			<td><img src="pic11.gif"></td>
			<td><img src="pic11.gif" class=mask></td>
		</tr>
	</table>
</body>
</html>
```

运行程序，效果如图 11-59 所示。



图 11-59　遮罩效果

11.11.11　波形滤镜——wave

波形滤镜能造成一种强烈的变形幻觉，它对目标对象生成正弦波变形。实际上，它是把对象按垂直的波形样式打乱。

语法：filter:wave(add=参数值, freq=参数值, lightstrength=参数值, phase=参数值,strength=参数值)

说明：在该滤镜的参数中，add 表示是否要把对象按照波形样式打乱，默认是 true；freq 为波纹的频率，也就是指定在对象上一共需要产生多少个完整的波纹；lightstrength 参数设定对于波纹增强光影的效果，范围是 0~100；phase 参数用来设置正弦波的偏移量；strength 用于设定振幅。

实例代码：

```
<html>
<head>
<title>产生波形效果</title>
<style type="text/css">
<!--
	body{ font-size:12pt}
	h2{ font-family: "黑体"; font-size:15pt; }
	.wave{ filter: wave(add=false, freq=2,lightstrength=20,phase=5,strength=25)}
	.wave2{filter: wave(add=true, freq=3,lightstrength=180,phase=180,strength=80)}
-->
</style>
</head>
<body>
	<h2>下面是对同样的图像使用了波形滤镜</h2>
	<table border=1 bordercolor="#660000" cellpadding=5 width=660 align="center">
		<tr height=30 align="center">
			<td width=220>原图</td>
			<td width=220>产生波浪的效果</td>
			<td width=220>不同设置的波浪效果</td>
		</tr>
		<tr height=260 align="center">
			<td><img src="pic09.jpg"></td>
			<td><img src="pic09.jpg" class=wave></td>
			<td><img src="pic09.jpg" class=wave2></td>
		</tr>
	</table>
</body>
</html>
```

运行程序，效果如图 11-60 所示。



图 11-60　波形效果

第12章 使用 JavaScript 脚本

- JavaScript 简介
- JavaScript 的基本语法
- JavaScript 对象
- JavaScript 事件
- 利用 JavaScript 制作特效

前面介绍了在页面中插入表单的方法，但是插入表单只是实现了交互功能中收集信息的功能，它并没有真正处理收集的这些信息。如果要处理这些信息，往往要用到脚本语言。在 HTML 中，最常见的脚本语言就是 JavaScript。

12.1　JavaScript 简介

12.1.1　什么是 JavaScript

JavaScript 是一种新的描述语言，可以被嵌入 HTML 文件之中。它是一种基于对象和事件驱动，并具有安全性能的脚本语言。使用它的目的是与 HTML 超文本标记语言、Java 脚本语言一起实现在一个 Web 页面中链接多个对象，与 Web 客户交互作用，从而可以开发客户端的应用程序等。

透过 JavaScript 可以回应使用者的需求事件（如：form 的输入）而不需要通过网络来回传输资料。简单地说，当一位使用者输入一项信息时，不需要先将信息传给服务器处理，再传回来，而是可以直接被客户端的程序处理。

12.1.2　JavaScript 特点

JavaScript 的出现弥补了 HTML 语言的缺陷，它是 Java 与 HTML 折衷的选择，具有以下几个特点：

- ❑ JavaScript是一种脚本编写语言，采用小程序段的方式实现编程，也是一种解释性语言，提供了一个简易的开发过程。它与HTML标识结合在一起，从而方便用户的使用操作。
- ❑ JavaScript是一种基于对象的语言，同时也可以看作是一种面向对象的语言。这意味着它能运用自己已经创建的对象，因此许多功能可以来自于脚本环境中对象的方法与脚本的相互作用。
- ❑ JavaScript具有简单性。首先它是一种基于Java基本语句和控制流之上的简单而紧凑的设计，其次它的变量类型采用弱类型，并未使用严格的数据类型。
- ❑ JavaScript是一种安全性语言，它不允许访问本地硬盘，并且不能将数据存入到服务器上，不允许对网络文档进行修改和删除，只能通过浏览器实现信息浏览或动态交互，从而有效地防止数据的丢失。
- ❑ JavaScript是动态的，它可以直接对用户或客户输入作出响应，无须经过Web服务程序。它对用户的反映响应，是采用以事件驱动的方式进行的。所谓事件驱动，就是指在主页中执行了某种操作所产生了动作，从而触发相应的事件响应。
- ❑ JavaScript具有跨平台性。它依赖于浏览器本身，与操作环境无关，只要能运行浏览器并支持JavaScript浏览器的计算机就能正确执行。

12.1.3　一个简单的 JavaScript 实例

在 HTML 中插入 JavaScript 代码的基本语法如下：

```
<Script Language ="JavaScript">
    JavaScript 语言代码;
    JavaScript 语言代码;
        ……
</Script>
```

说明：通过标识<Script>、</Script>指明在标记中嵌入的是脚本语言；通过设置 Language 属性的值为 JavaScript 说明标识中使用的是何种语言。

下面创建第一个 JavaScript 的小实例，实例代码如下：

```
<html>
<head>
<title>将 JavaScript 语言脚本嵌入 HTML 中</title>
<Script Language="JavaScript">
    <!--
        document.write("第一个 JavaScript 的小实例");
    -->
</Script>
</head>
<body>
<Script Language="JavaScript">
    <!--
        alert("弹出一个小窗口")
    -->
</Script>
</body>
</html>
```

运行这段代码，可以看到如图 12-1 所示的效果。由这一实例可以看出，<Script>标记可以放在 HTML 文件中的任何地方，包括头部和主体等。



图 12-1　程序运行效果

12.2　JavaScript 的基本语法

JavaScript 脚本语言同其他语言一样，有基本的数据类型、表达式和算术运算符以及程序的基本框架结构。JavaScript 提供了 4 种基本的数据类型用来处理数字和文字，而变量是用于提供存放信息的地方，表达式则可以完成较复杂的信息处理。

12.2.1　常量

JavaScript 的常量通常又称字面常量，它是不能改变的数据，主要包括如下几种：

- 整型常量：可以使用十六进制、八进制和十进制表示其值。
- 实型常量：由整数部分加小数部分表示，如12.32、193.98。可以使用科学或标准方法表示，如5E7、4e5等。
- 布尔值：只有两种状态，true或false。主要用来说明或代表一种状态，从而判断下一步的操作。
- 字符型常量：使用单引号（'）或双引号（"）引起来的一个或几个字符。如"This is JavaScript"、'3245'、"ew231"等。
- 空值：在JavaScript语言中有一个空值null，表示什么也没有。如果试图引用没有定义的变量，就会返回一个null值。
- 特殊字符：JavaScript中有些以反斜杠（/）开头的不可显示的特殊字符，通常称为控制字符。

注意： 在 JavaScript 中是区分大小写的，例如，其中的布尔值 true 和 false 要使用小写，如果写成 True 或 False 是错误的。

12.2.2　变量

变量的主要作用是存取数据、提供存放信息的容器。变量可分为全局变量和局部变量，通常声明于函数（Function）内的都属于局部变量，在 Script 标记内及函数外的则属于全局变量，局部变量只有在函数内可以存取，而全局变量则不受限制。

给 JavaScript 中的变量命名要注意以下几点：

- 必须是一个有效的变量，即变量以字母开头，中间可以出现数字，如test1、text2等。除下划线（_）作为连字符外，变量名称不能有空格、加号（+）、减号（-）、逗号（,）或其他符号。
- 不能使用JavaScript中的关键字作为变量。在JavaScript中定义了40多个关键字（如var、int、true等）供JavaScript内部使用，不能作为变量名称。
- 在对变量命名时，最好把变量的意义与其代表的意思对应起来，以免出现错误。

在JavaScript中，变量可以使用var关键字在使用前先作声明，并可赋值。例如：

```
var myname;
```

此处定义了一个名为 myname 的变量，但没有赋值。

```
var myname="John Sanic";
```

此处定义了一个名为 myname 的变量，同时赋予其值为 John Scnic。

变量也可以不作声明，在使用时再根据数据类型直接确定变量类型，如：

```
x=100;
y="125";
tip=true;
```

其中 x 为整数，y 为字符串，tip 为布尔型。

在为变量赋值时，可以添加 HTML 中的标记语言，下面通过实例说明变量的使用方法，实例代码如下：

```
<html>
<head>
<title>使用变量</title>
<style>
    <!--
        h2{font-family: "黑体"; font-size: 16pt}
        body{font-size: 12pt}
    -->
</style>
<Script language="JavaScript">
    <!--
        examword="欢迎学习 JavaScript";
        secword="<p>通过 JavaScript 脚本语言，能让页面效果变得更加绚丽多彩</p>"
    -->
</Script>
</head>
<body>
    <h2>下面显示的是定义的变量值：</h2>
<Script Language="JavaScript">
    <!--
        document.write(examword);
        document.write(secword);
    -->
</Script>
</body>
</html>
```

运行程序，效果如图 12-2 所示。可以看出，在页面中成功地显示了前面使用 JavaScript 定义的两个变量的值。

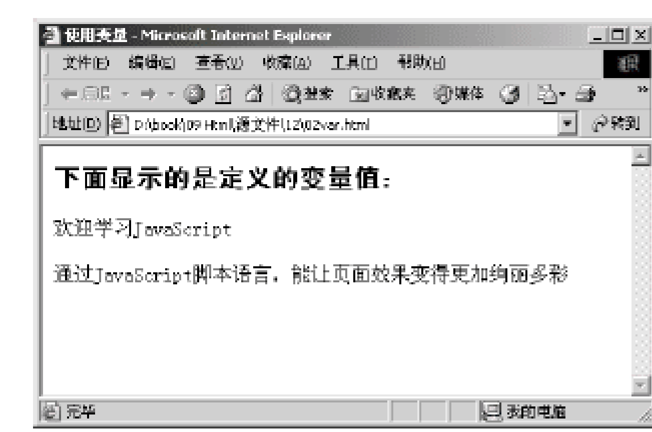


图 12-2　使用变量的效果

12.2.3　运算符

运算符是用于完成操作的一系列符号，在 JavaScript 中包括算术运算符（如+、−、*、/等）、比较运算符（如!=、==等）、逻辑布尔运算符（如！）以及字符串运算符（如+=）。

在 JavaScript 中主要有双目运算符和单目运算符。双目运算符的基本写法如下：

操作数 1 运算符 操作数 2

可以看出双目运算符由两个操作数和一个运算符组成，如 50−40、"this"+"that"等。而单目运算符只需一个操作数，其运算符可在前或后，如++1。

- ❑ 算术运算符：JavaScript中的算术运算符有单目运算符和双目运算符。其中，双目运算符主要包括+（加）、-（减）、*（乘）、/（除）、%（取模）。单目运算符主要包括−（取反）、~（取补）、++（递加1）、--（递减1）。

- ❑ 比较运算符：比较运算的基本操作过程是首先对其操作数进行比较，再返回一个true或false值。在JavaScript中的运算符主要包括<（小于）、>（大于）、<=（小于等于）、>=（大于等于）、==（等于）、!=（不等于）。
- ❑ 逻辑运算符：逻辑运算符也称为布尔运算符，主要包括!（取反）、&&（逻辑与）、||（逻辑或）。

实例代码：

```
<html>
<head>
<title>使用 JavaScript 进行计算</title>
<style>
	<!--
		h2{font-family: "黑体"; font-size: 16pt}
		body{font-size: 12pt}
	-->
</style>
</head>
<body>
	<h2>下面显示各种类型的运算结果：</h2>
<Script Language="JavaScript">
	<!--
		document.write("算术运算中的加法 5+7 的结果：");
		document.write(5+7);
		document.write("<br>");
		document.write("比较运算 35>=36 的结果是：");
		document.write(35>=36);
		document.write("<br>");
		document.write("逻辑运算 true&&false 的值：");
		document.write(true&&false);
	-->
</Script>
</body>
</html>
```

运行程序，效果如图 12-3 所示。

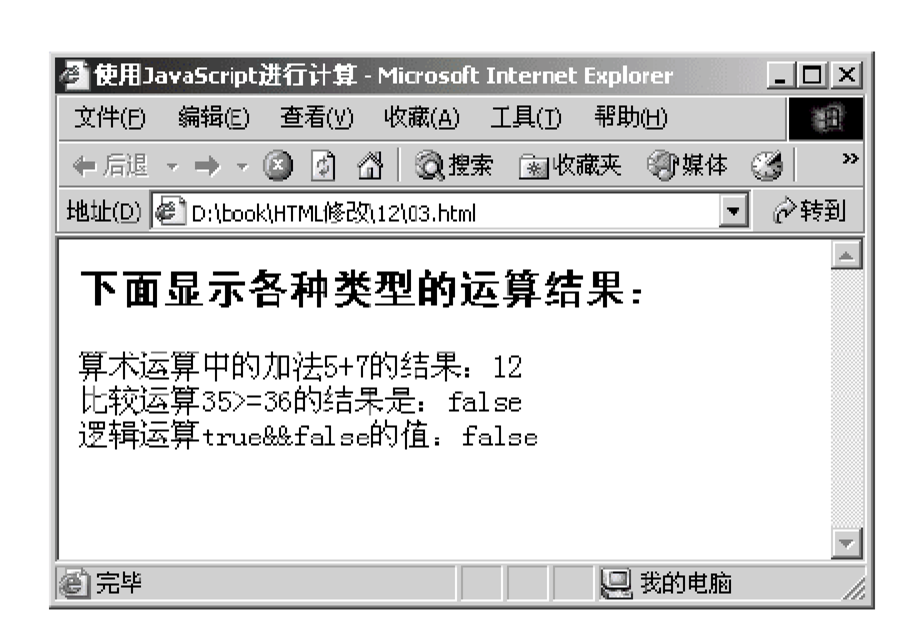


图 12-3　使用运算符进行计算

12.2.4　表达式

在定义完变量后，就可以对它们进行赋值、改变、计算等一系列操作，完成这一过程的操作就需要使用表达式。可以说，表达式就是变量、常量、布尔及运算符的集合，因此表达式可以分为算术表述式、字符表达式、赋值表达式以及布尔表达式等。

此外还有一种表达式称为条件表达式，它通过判断一个条件是否符合来设置相应的显示结果，表达式的写法如下：

(条件)？结果 1：结果 2

若条件的结果值为真，则表述式的结果为 1，否则为 2。
实例代码：

```
<html>
<head>
<title>使用表达式</title>
<style>
	<!--
		h2{font-family: "黑体"; font-size: 16pt}
		body{font-size: 12pt}
	-->
</style>
</head>
<body>
	<h2>下面应用 JavaScript 中的表达式：</h2>
<Script Language="JavaScript">
	<!--
	exam1=(3+6>8)?"yes":"no";
	exam2=(3+6<8)?"yes":"no";
	document.write("判断 3+6>8 是否正确，如果正确输出 yes，否则输入 no：<br>");
	document.write(exam1);
	document.write("<br><br>");
	document.write("判断 3+6<8 是否正确，如果正确输出 yes，否则输入 no：<br>");
	document.write(exam2);
	-->
</Script>
</body>
</html>
```

运行程序，效果如图 12-4 所示。

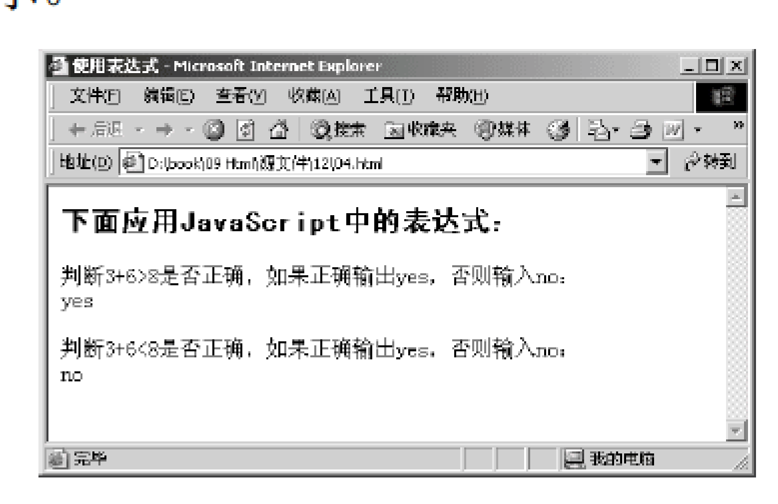


图 12-4　使用表达式的效果

12.2.5　基本语句

在任何一种语言中，程序控制流是必须的，它能使得整个程序减小混乱，使之顺利按一定的方式执行。下面是 JavaScript 常用的程序控制流语句。

1．if-else 条件语句

if-else 语句是 JavaScript 中最基本的控制语句，通过它可以改变语句的执行顺序。表达式中必须使用关系语句来实现判断，它作为一个布尔值来计算，可以将零和非零的数分别转化成 false 和 true。

基本格式如下：

```
if(表述式){
    执行语句 1;
        ……
    }
else{
    执行语句 2;
        ……
    }
```

说明：当表达式的值为 true，则执行语句 1，否则执行语句 2。若 if 后的语句有多行，则必须使用花括号将其括起来。

if 语句可以嵌套使用，从而实现更加复杂的判断，其语法如下：

```
if(布尔值或布尔表达式)语句 1;
else(布尔值或布尔表达式)语句 2;
else if(布尔值或布尔表达式)语句 3;
……
else 语句 4;
```

说明：在这种情况下，每一级的布尔表述式都会被计算，若某一级布尔表达式为真，则执行其后面相应的语句，否则向下判断。如果一直到最后的布尔表达式都为假，则执行 else 后的语句。

实例代码：

```
<html>
<head>
<title>使用 if-else 条件语句</title>
<style>
    <!--
        h2{font-family: "黑体"; font-size: 16pt}
        body{font-size: 12pt}
    -->
</style>
</head>
<body>
    <h2>标准体重的测试</h2>
    标准体重（公斤）＝身高（厘米）－105<br><br>
    假如体重为 66 公斤，身高 169 厘米，那么这个人的体重是否超重呢？<br>
<Script Language="JavaScript">
    <!--
        weight=66;
        height=169;
        standard=height-105;
        judge=(weight-standard)*100/standard;
        document.write("计算结果：体重超出标准体重的百分比为");
        document.write(judge);
        if (judge>20)
```

```
            document.write("<br>偏胖，要注意减肥");
          else if (judge<-20)
            document.write("<br>偏瘦，注意增加营养");
          else
            document.write("<br>体型标准，注意保持");
     -->
</Script>
</body>
</html>
```

运行程序，效果如图 12-5 所示。

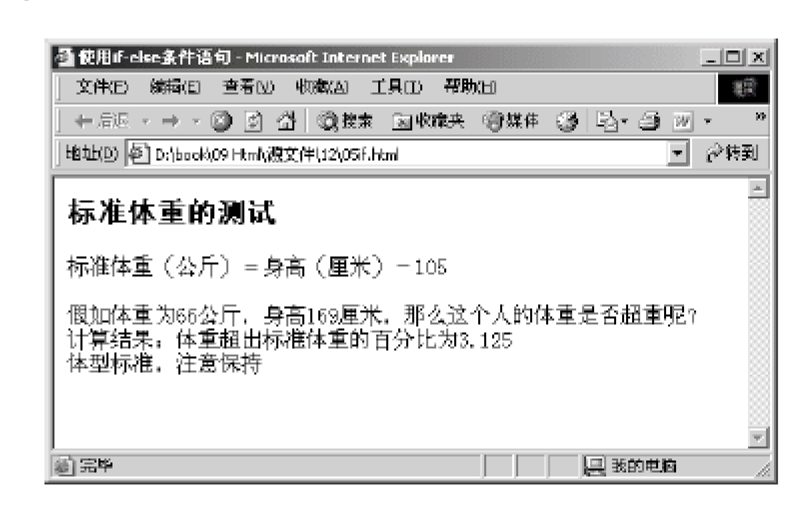


图 12-5　使用 if-else 语句判断

2．for 循环语句

for 语句用于实现条件循环，即当条件成立时，执行语句集，否则跳出循环体。

基本格式：

```
for(初始化;条件;增量)
{
    语句集;
    ……
}
```

说明：初始化参数告诉循环的开始位置，必须赋予变量初值；条件是用于判别循环停止时的条件，若条件满足，则执行循环体，否则跳出循环；增量主要定义循环控制变量在每次循环时按什么方式变化。在 3 个主要语句之间，必须使用分号（;）分隔。

例如：

```
for (i = 0; i < 10; i++) {
    x[i]= i;
}
```

说明：初始值 i=0，条件 i<10（也就是从 0~9）；i++表示 i=i+1，也就是递增值为 1。这段代码表示从 0 开始每次递增 1 给数组 x[i]赋值，一直到 i 为 10 跳出循环。

实例代码：

```
<html>
<head>
<title>使用 for 语句计算累积的和</title>
<style>
```

```
    <!--
        body{font-size: 12pt}
    -->
</style>
</head>
<body>
    计算从 1 到 10 的和：
    <Script Language="JavaScript">
    <!--
        sum=0;
        for(i=1;i<11;i++)
          sum=sum+i
          document.write(sum);
    -->
    </Script>
</body>
</html>
```



图 12-6　使用 for 语句计算和

运行程序，效果如图 12-6 所示。

3．while 循环

该语句与 for 语句一样，当条件为真时，重复循环，否则退出循环。

基本格式：

```
while(条件){
    语句集;
    ……
}
```

说明：在 while 语句中，条件语句只有一个，当条件不符合时跳出循环。

它与 for 语句的主要区别在于：使用 for 语句在处理有关数字时更易看懂，也较紧凑；而 while 循环对复杂的语句效果更特别。

4．break 语句

使用 break 语句可以使循环从 for 或 while 中跳出。

基本语法：break;

说明：当程序遇到 break 语句时，会跳出循环并执行下一条语句。

实例代码：

```
<html>
<head>
<title>使用 break 语句跳出循环</title>
<style>
    <!--
        body{font-size: 12pt}
    -->
</style>
</head>
<body>
```

```
    从 1 开始输出整数，当循环到能被 7 整除时跳出循环，否则输出到 20<br><br>
    <Script Language="JavaScript">
    <!--
    for(i=1;i<21;i++){
          if(i%7==0)
            break;
          document.write(i+"  ")
      }
    -->
    </Script>
</body>
</html>
```

运行程序，效果如图 12-7 所示，i=7 时符合跳出条件，因此不再输出，程序只输出 1 到 6 的数字。

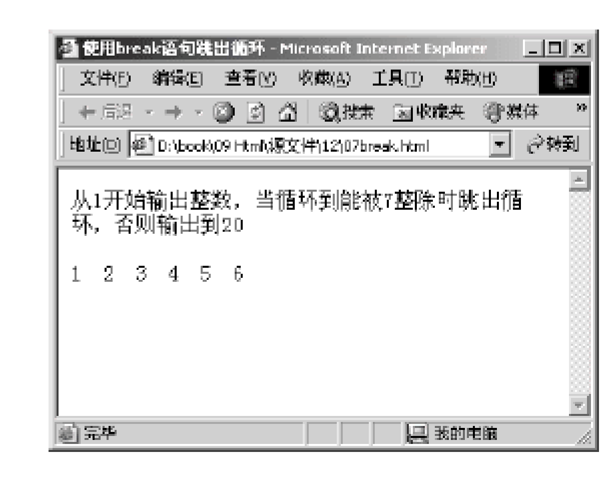


图 12-7　使用 break 语句跳出循环

5．continue 语句

使用 continue 语句则使程序跳过循环内剩余的语句而进入下一次循环。

基本语法：continue;

将上一个实例中的 break 语句更改为 continue 语句，运行实例的效果如图 12-8 所示。

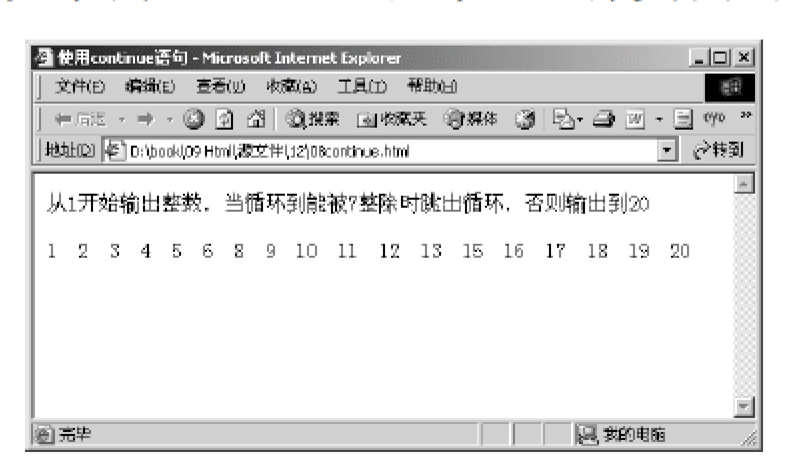


图 12-8　使用 continue 语句结束当前一次的循环

由图 12-8 可以看出，当遇到 continue 语句时并不跳出整个循环，只是结束当前的这一次循环，因此当 i=7 和 i=14 时并不跳出循环，而是进入下一轮循环中，因此输出的是除去能被 7 整除的 7 和 14 以外的从 1 到 20 的数字。

6．switch 语句

当判断条件比较多时，为了使程序更加清晰，可以使用 switch 语句。使用 switch 语句时，表达式的值将与每个 case 语句中的常量作比较。如果相匹配，则执行该 case 语句后的代码；如果没有一个 case 的常量与表达式的值相匹配，则执行 default 语句。当然，default 语句是可选的。如果没有相匹配的 case 语句，也没有 default 语句，则什么也不执行。

基本语法：

```
switch (表达式) {
case  值 1:
      语句 1;
      break;
case  值 2:
```

```
        语句 2;
        break;
    ……
    default:
        语句 N;
}
```

下面通过一个实例说明 switch 语句的使用方法。给定一组成绩，判断成绩的等级。如果小于 60 分则为不及格；60~69 分为及格；70~79 分为普通；80~89 分为较好，90 分以上则设置为优秀。实例代码如下：

```
<html>
<head>
<title>使用 switch 语句</title>
<style>
    <!--
        body{font-size: 12pt}
    -->
</style>
</head>
<body>
判断下面同学成绩的等级：<br>
75  88  56  99  62  <br><br>
    <Script Language="JavaScript">
    <!--
    var mark= new Array();
    mark[1]=75;
    mark[2]=88;
    mark[3]=56;
    mark[4]=99;
    mark[5]=62;
    for (i = 1; i < 6; i++) {
      switch(parseInt(mark[i]/10)){
        case 0:
         val="不及格";
         break;
        case 1:
         val="不及格";
         break;
        case 2:
         val="不及格";
         break;
        case 3:
         val="不及格";
         break;
        case 4:
         val="不及格";
         break;
```

```
            case 5:
             val="不及格";
             break;
            case 6:
             val="及格";
             break;
            case 7:
             val="普通";
             break;
            case 8:
             val="较好";
             break;
            case 9:
             val="优秀";
             break;
            case 10:
             val="优秀";
             break;
          }
        document.write(mark[i]+" :  "+val+"<br>")
        }
        -->
        </Script>
    </body>
    </html>
```

在该实例中，通过 var 定义了数组变量 mark[i]，通过 parseInt()函数对数组的值进行取整，然后判断它的值，如果小于 6 则说明数组的值小于 60，因此设置为“不及格”，依此类推。运行程序，效果如图 12-9 所示。

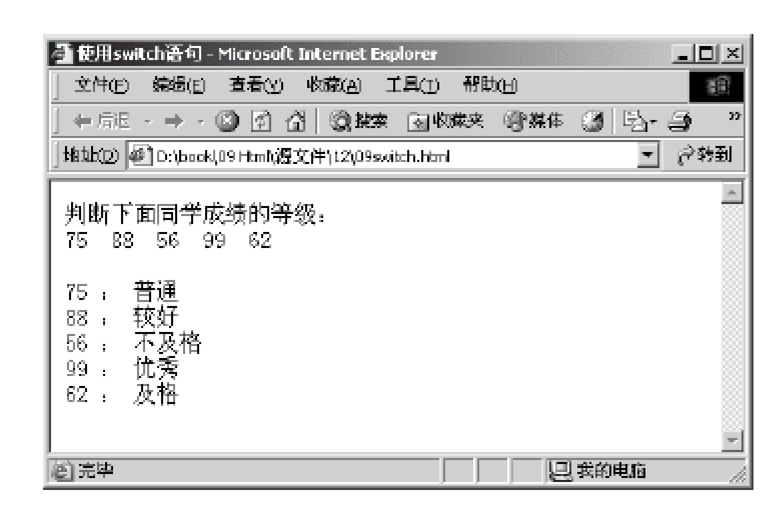


图 12-9　使用 switch 语句判断成绩等级

注意： *在该实例中使用了 new()创建数组；使用了 parseInt()方法进行取整。*

12.2.6　函数

通常在进行一个复杂的程序设计时，总是根据所要完成的功能，将程序划分为一些相对独立的部分，每部分编写一个函数，从而使各部分充分独立，任务单一，程序清晰，易懂、易读、易维护。JavaScript 函数由关键字 function 定义，通过指定函数名（实参）来调用一个函数。函数可以封装在程序中可能

要重复使用的模块中，并可作为事件驱动的结果而调用。关于事件，在后面的内容中将作具体讲解。

在 JavaScript 中，函数定义的方法如下：

```
function 函数名 (参数,变量){
    函数体;
    return 表达式;
}
```

说明：在这一语法中，函数名用于定义函数名称；参数是传递给函数使用或操作的值，其值可以是常量、变量或其他表达式；return 则用于设定函数的返回值。

在 JavaScript 中，函数名对大小写是敏感的，也就是 JavaScript 中的函数名区分大小写，如 New 和 new 是不同的两个函数。

12.3　JavaScript 对象

JavaScript 可以根据需要创建自己的对象，从而进一步扩大其应用范围。JavaScript 中的对象是由属性和方法两个基本元素构成的。前者是对象在实施其所需要行为的过程中，实现信息的装载单位，从而与变量相关联；后者是指对象能够按照设计者的意图而被执行，从而与特定的函数关联。

12.3.1　对象属性的引用

对于对象的属性，可以通过 3 种方法进行引用。下面举例说明，假如 city 是一个已经存在的对象，而 name 和 date 则是它的两个属性，如果要对这两个属性赋值，可以采用如下 3 种方法：

- 使用点运算符：
```
city.name="宁波";
```
- city.date="1998";
- 通过对象的下标实现引用：通过数组形式的访问属性，可以使用循环操作获取其值。
```
city[0]= "宁波";
city[1]= "1998";
function show(object){
for (j=0;j<2; j++)
  document.write(object[j])
  }
```
- 通过字符串的形式实现：
```
city["name"]="宁波";
city["date"]="1998";
```

12.3.2　对象方法的使用

在对象中除了使用属性外，有时还需要使用方法。在 JavaScript 中对象方法的引用非常简单。在对象的定义中使用了 This.meth=FunctionName 语句，实质上对象的方法就是一个函数 FunctionName。

其语法：

对象名.方法名()=函数名()

然后再定义函数的具体功能即可实现对象方法的定义。

如果要直接调用已知对象的方法，可以直接使用点运算符。例如要调用 math 对象中的 cos()方法，可以使用如下代码：

```
document.write(math.cos(35))
document.write(math.sin(80))
```

12.3.3　浏览器的内部对象

使用浏览器的内部对象系统，可实现与 HTML 文档进行交互。它的作用是将相关元素组织包装起来，提供给程序设计人员使用，从而减轻编程人员的劳动，提高设计 Web 页面的能力。浏览器的内部对象主要包括以下几个：

- 浏览器对象（navigator）：提供有关浏览器的信息。
- 文档对象（document）：document对象包含了与文档元素（elements）一起工作的对象，它将这些元素封装起来供编程人员使用。
- 窗口对象（windows）：window对象处于对象层次的最顶端，它提供了处理navigator窗口的方法和属性。
- 位置对象（location）：location对象提供了与当前打开的URL一起工作的方法和属性，它是一个静态的对象。
- 历史对象（history）：history对象提供了与历史清单有关的信息。

利用这些对象可以对浏览器环境中的事件进行控制和处理。在 JavaScript 中提供了非常丰富的内部方法和属性，从而减轻了编程人员的工作，提高了编程效率。在这些对象系统中，文档对象属于非常重要的，它位于最底层，但对实现 Web 页面信息交互起着关键作用，因而它是对象系统的核心部分，下面具体介绍这些对象的常用属性和方法。

1. navigator 对象

navigator 对象可用来存取浏览器的相关信息，其常用属性见表 12-1。

表12-1　navigator对象的常用属性

属　　性	说　　明
appName	浏览器的名称
appVersion	浏览器的版本
appCodeName	浏览器的代码名称

续表

属　　性	说　　明
browserLanguage	浏览器所使用的语言。如zh-cn表示中文、en表示英文等
plugins	可以使用的插件信息
platform	浏览器系统所使用的平台，如Win32等
cookieEnabled	浏览器的cookie功能是否打开

实例代码：

```
<html>
<head>
<title>调用 navigator 对象的属性</title>
<style>
    <!--
        body{font-size: 12pt}
    -->
</style>
</head>
<body>
测试您所使用的浏览器名称、版本及系统平台：<br><br>
    <Script Language="JavaScript">
    <!--
    document.write("您所使用的浏览器名称为："+navigator.appName+"<br>")
    document.write("浏览器所使用的语言为："+navigator.browserLanguage+"<br>")
    document.write("您所使用的浏览器版本为："+navigator.appVersion+"<br>")
    -->
    </Script>
</body>
</html>
```

运行程序，效果如图 12-10 所示。

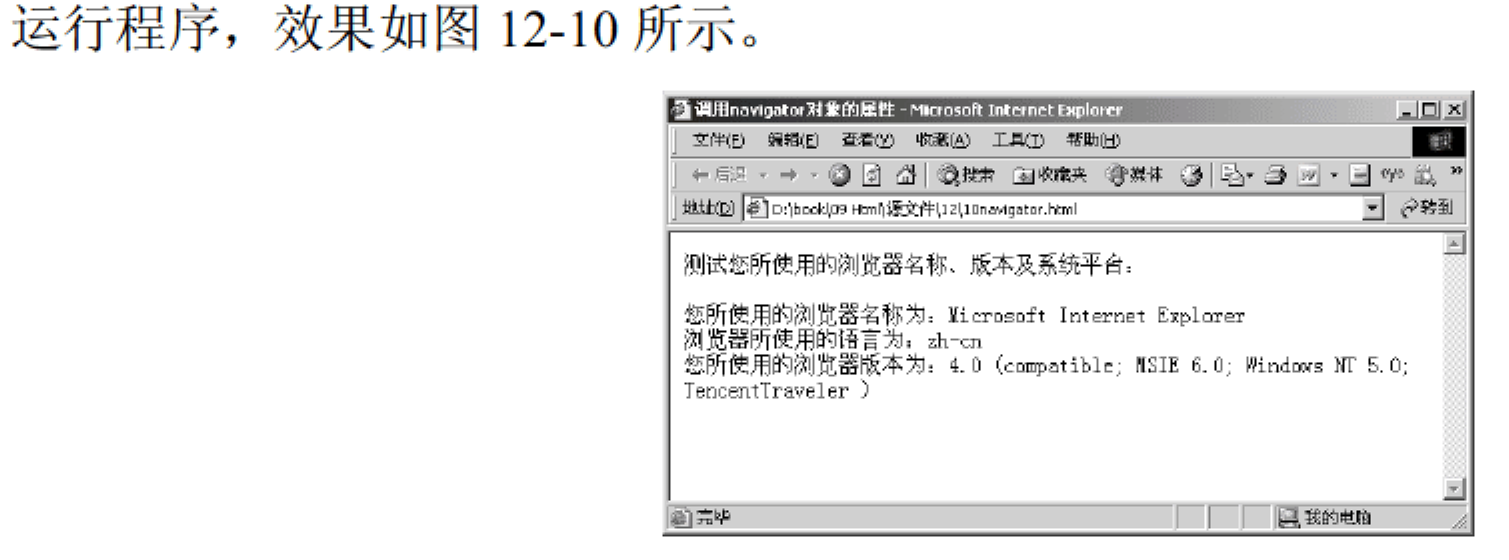


图 12-10　使用 navigator 对象

2．document 对象

JavaScript 是基于对象的脚本编程语言，它的输入输出是通过对象来完成的，其中输出可通过 document 对象实现。在 document 中主要有 links、anchor 和 form 3 个最重要的对象。

- anchor锚对象：它是指<A name=…> </A>标识在HTML源码中存在时产生的对象，它包含着文档中所有的anchor信息。
- links链接对象：是指用<A href=…> </A>标记链接一个超文本或超媒体的元素作为一个特定

的URL。

- form窗体对象：是文档对象的一个元素，它含有多种格式的对象储存信息，使用它可以在JavaScript脚本中编写程序进行文字输入，并可以用来动态改变文档的行为。通过document.Forms[]数组来使得在同一个页面上可以有多个相同的窗体，使用forms[]数组要比使用窗体名字方便得多。

document 对象有以下方法：

- 输出显示write()和writeln()：该方法主要用来实现在Web页面上显示输出信息。

注意： writeln()与 write()的惟一不同之处在于在末尾加了一个换行符；为了正常显示其输出信息，必须指明<pre> </pre>标记，使之告诉编辑器；输出的文档类型，可以由浏览器中有效的合法文本类型所确定。

- 关闭文档流close()：在多个文档对象中，必须使用close()来关闭一个对象后，才能打开另一个文档对象。
- 清除文档内容clear()：用于清除文档中的内容。

实例代码：

```
<html>
<head>
<title>使用 document 对象</title>
</head>
<body>
<form name="mytable">
请输入数据：
<Input Type="text" Name="text1" Value="">
</form>
<A name="Link1" href="02var.htm">链接到第一个文本</a>   
<A href="03.htm">链接到第二个文本</a>   
<A href="04.htm">链接到第三个文本</a><br><br>
<Script Language="JavaScript">
document.write("本页面有"+document.links.length+"个链接"+"  ");
document.write("有"+document.anchors.length+"个锚"+"  ");
document.write("面有"+document.forms.length+"个窗体");
</Script>
</body>
</html>
```

运行程序，效果如图 12-11 所示。

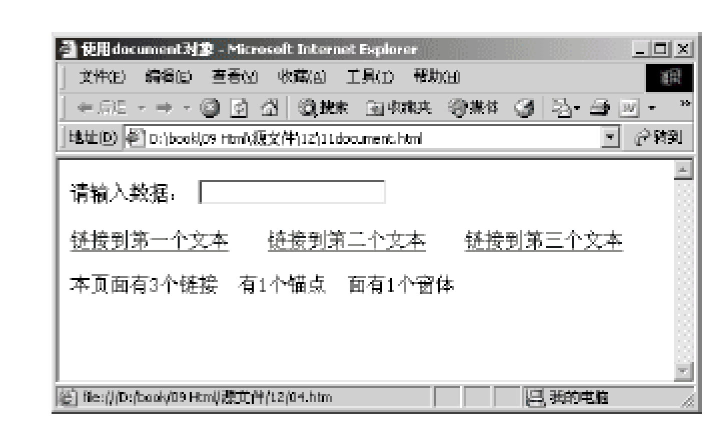


图 12-11　使用 document 对象

3．window 对象

window 对象处于对象层次的最顶端，它提供了处理 navigator 窗口的方法和属性。JavaScript 的输出可通过 document 对象的方法来完成，而输入则可以通过 window 对象来实现。window 对象常用的方法主要见表 12-2。

表12-2　window对象的常用方法

方　法	方法的含义及参数说明
open(url,windowName,parameterlist)	创建一个新窗口，3个参数分别用于设置URL地址、窗口名称和窗口打开属性（一般可以包括宽度、高度、定位、工具栏等）
close()	关闭一个窗口
alert(text)	弹出式窗口，text参数为窗口中显示的文字
confirm(text)	弹出确认域，text参数为窗口中的文字
promt(text,defaulttext)	弹出提示框，text为窗口中的文字，defaulttext参数用来设置默认情况下显示的文字
moveBy(水平位移,垂直位移)	将窗口移至指定的位移
moveTo(x,y)	将窗口移动到指定的坐标
resizeBy(水平位移,垂直位移)	按给定的位移量重新设定窗口大小
resizeTo(x,y)	将窗口设定为指定大小
back()	页面的后退
forward()	页面前进
home()	返回主页
stop()	停止装载网页
print()	打印网页
status	状态栏信息
location	当前窗口URL信息

下面以一个实例来说明 window 对象的功能，实例代码如下：

```
<html>
<head>
<title>使用 window 对象</title>
</head>
<body>
在该页面将会依次弹出确认窗口以及一个新的页面。
<Script Language="JavaScript">
      <!--
      confirm("确定要显示页面内容么？");
      window.open("exam.html","newwindow","width=450,height=220,toolbar=no,menubar=no,scrollbars=yes");
      -->
</Script>
</body>
</html>
```

运行程序，效果如图 12-12 所示。

单击“确定”按钮会继续打开页面的内容，即弹出新的页面，如图 12-13 所示。



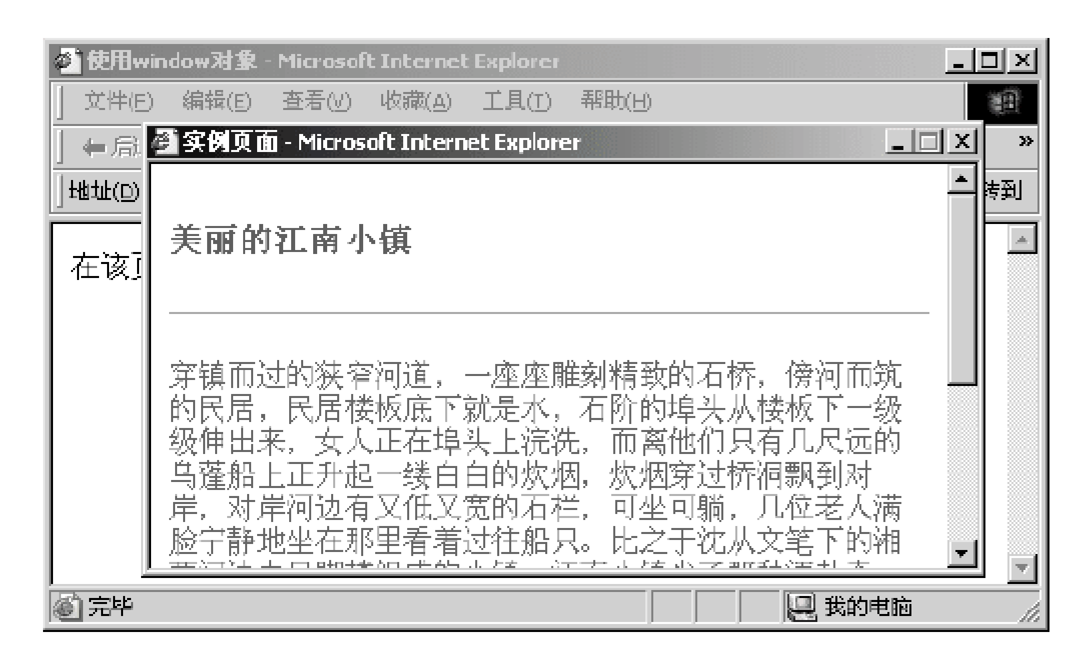



图 12-12　使用 window 对象弹出确认窗口

图 12-13　新打开一个页面窗口

4．location 对象

location 对象是一个静态的对象，它描述的是某一个窗口对象所打开的地址。要表示当前窗口的地址，只需要使用 location 即可；若要表示某一个窗口的地址，就使用“窗口对象.location”。

注意：属于不同协议或不同主机的两个地址之间不能互相引用对方的 location 对象，这是出于安全性的需要。例如，当前窗口打开的是地址为 a1.html 的下面的某一页，另外一个窗口（对象名为：winb）打开的是 b2.html 网页。如果在当前窗口使用 winb.location，就会出现出错信息：“没有权限”。

location 对象常用的属性见表 12-3。

表12-3　常用的location属性

属　　性	实现的功能
protocol	返回地址的协议，取值为'http:'、'https:'、'file:'等
hostname	返回地址的主机名，例如“http://www.microsoft.com/china/”的地址主机名为www.microsoft.com
port	返回地址的端口号，一般http的端口号是80
host	返回主机名和端口号，如www.a.com:8080
pathname	返回路径名，如“http://www.a.com/b/c.html”的路径名为b/c.html
hash	返回“#”以及以后的内容，如地址为c.html#chapter4，则返回#chapter4；如果地址里没有“#”，则返回空字符串
search	返回“?”以及以后的内容；如果地址里没有“?”，则返回空字符串
href	返回整个地址，即返回在浏览器的地址栏上显示的内容

location 对象常用的方法主要包括：

- ❑ reload()：相当于Internet Explorer浏览器上的“刷新”功能。
- ❑ replace()：打开一个URL，并取代历史对象中当前位置的地址。用这个方法打开一个URL后，单击浏览器的“后退”按钮将不能返回到刚才的页面。

下面通过实例来介绍 location 对象的使用方法，实例代码如下：

```
<html>
<head>
```

```
<title>使用 location 对象</title>
</head>
<body>
使用 location 对象设置新链接窗口的路径：<br>
<form>
    <Input type="button" Value="打开新的页面" onclick="window.location.href='exam.html';">
</form>
<br><br>
    下面返回本页面地址的协议和路径名<br><br>
<Script Language="JavaScript">
    <!--
    document.write("本页地址的协议是：");
    document.write(location.protocol);
    document.write("<br>本页面地址的路径名是：");
    document.write(location.pathname);
    -->
</Script>
</body>
</html>
```

提示：onclick 是一个触发事件，表示单击鼠标时进行的处理程序，在后面的章节中还将细致讲解。

运行程序，效果如图 12-14 所示。

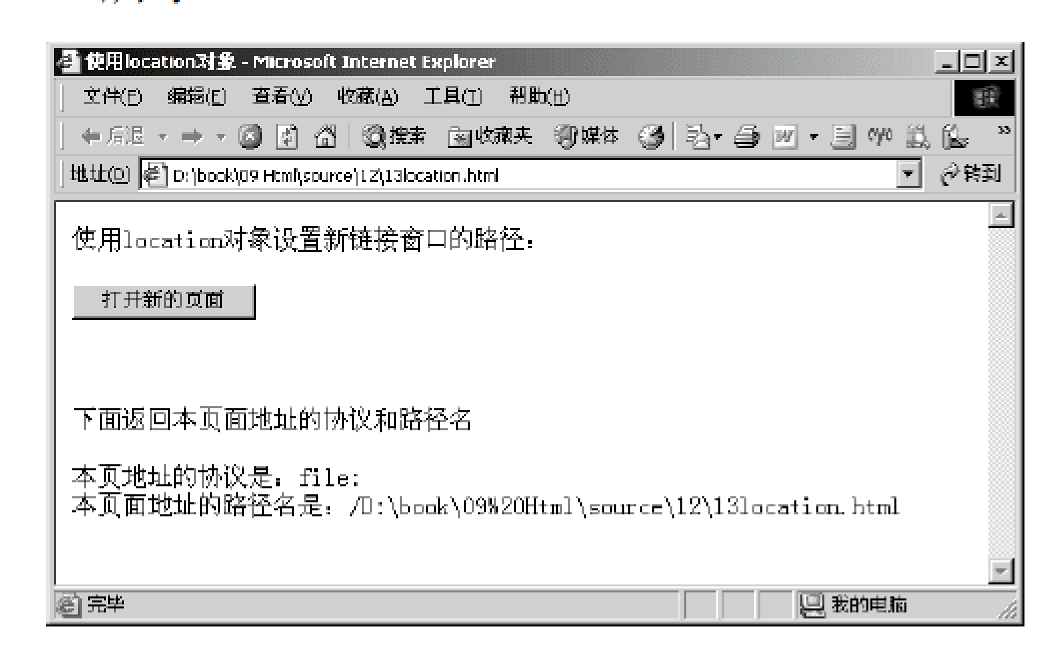


图 12-14　location 对象的使用方法

5．history 对象

history 对象是指浏览器的浏览历史，其最主要的属性就是 length，用于设定历史的项目数，也就是 JavaScript 历史中用浏览器的“前进”、“后退”按钮可以到达的范围。

history 对象常用的方法主要包括：

- back()：后退，与单击“后退”按钮是等效的。
- forward()：前进，与单击“前进”按钮是等效的。
- go()：history.go(x)实现在历史的范围内到达一个指定地址。如果x<0，则后退x个地址，如果x>0，则前进x个地址，如果x==0，则刷新现在打开的网页。

下面通过一个实例来说明 history 对象的使用方法，实例代码如下：

```
<html>
<head>
<title>使用 history 对象</title>
```

```
</head>
<body>
使用 history 对象访问浏览历史<br><br>
<form>
    <Input type="button" Value="后退到上一个历史记录" onclick="history.go(-1)"><br><br>
    <Input type="button" Value="前进到下一个历史记录" onclick="history.go(1)"><br><br>
    <Input type="button" Value="刷新本页面" onclick="history.go(0)">
</form>
</body>
</html>
```

运行程序，效果如图 12-15 所示。

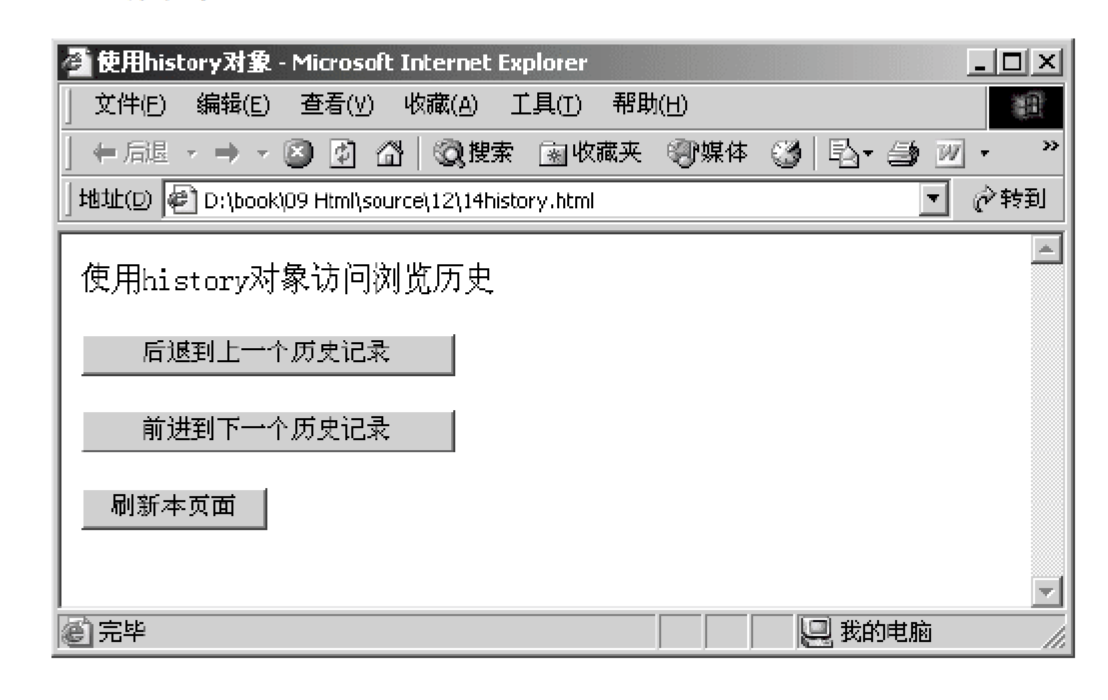


图 12-15　使用 history 对象访问历史记录

12.4　JavaScript 事件

12.4.1　事件简介

JavaScript 是基于对象的语言，而基于对象的基本特征就是采用事件驱动。通常鼠标或键盘的动作称之为事件，而由鼠标或键盘的动作引发的一连串程序动作，称之为事件驱动。对事件进行处理的程序或函数称为事件处理程序。在 JavaScript 中，对象的事件处理通常由函数实现。事件处理程序的语法与函数一样，因此也可以直接将函数作为事件处理程序。

事件处理程序的基本语法如下：

```
function 事件处理名(参数表){
事件处理语句集;
……
}
```

在调用事件处理程序时的基本语法如下：

```
事件驱动=处理程序
```

说明：在等号后，可以使用自己编写的函数作为事件处理程序，也可以使用 JavaScript 中内部的函数，还可以直接使用 JavaScript 的代码等。

JavaScript 事件驱动中的事件是通过鼠标或键盘动作引发的，主要包括单击事件（onClick）、改变事件（onChange）、选中事件（onSelect）、获得焦点事件（onFocus）、失去焦点事件（onBlur）、载入文件事件（onLoad）、卸载文件事件（onUnload）、鼠标覆盖事件（onMouseOver）、鼠标离开事件（onMouseOut）。下面对这些事件逐一进行讲解。

12.4.2　单击事件——onClick

onClick 事件是 JavaScript 中最常见的事件之一。用户单击鼠标按键时可产生 onClick 事件，同时 onClick 指定的事件处理程序或代码将被调用执行。

实例代码：

```
<html>
<head>
<title>单击事件</title>
</head>
<body>
使用 onClick 设置单击事件的处理程序<br>
<form>
     <Input type="button" Value="打开页面" onclick="window.open('exam.html','newwindow',
      'width=450,height=220,toolbar=no,menubar=no,scrollbars=yes');">
</form>
</body>
</html>
```

运行程序，单击页面中的“打开页面”按钮，可以打开一个名为 exam.html 的页面，如图 12-16 所示。

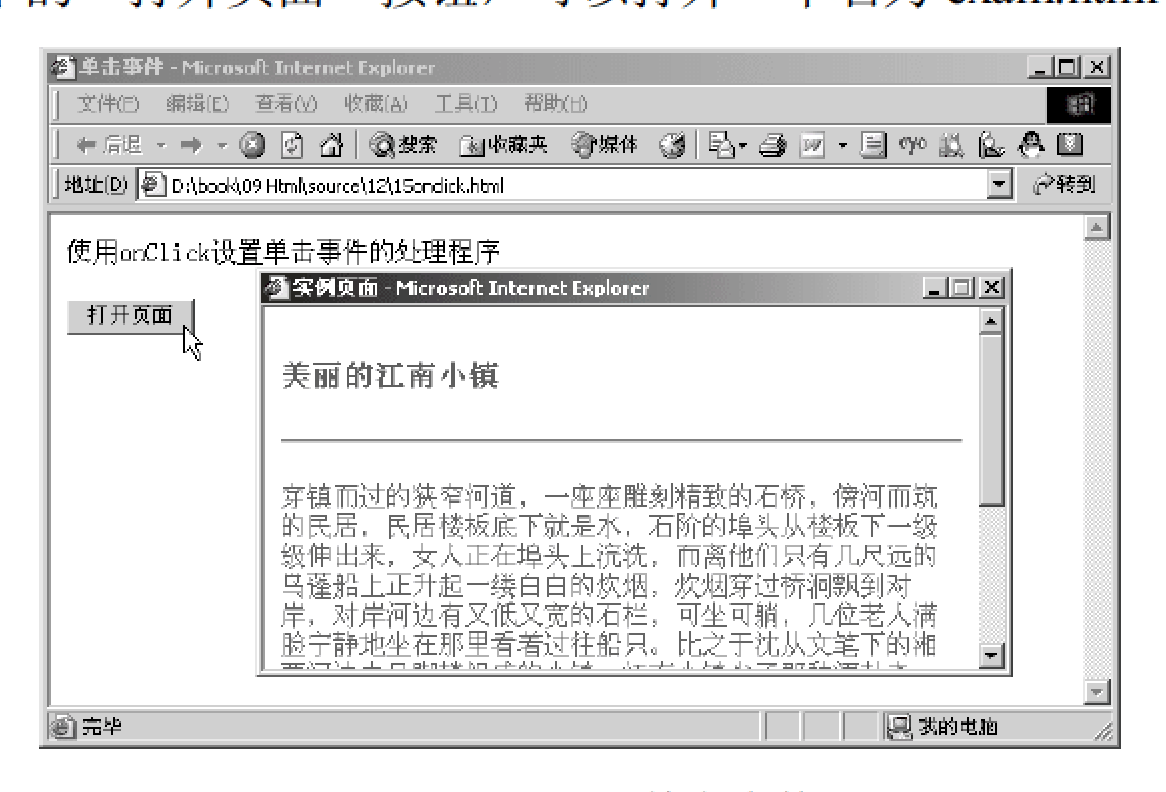


图 12-16　设置单击事件

12.4.3　改变事件——onChange

当 text 或 textarea 元素内的字符值改变或 Select 表格选项状态改变时发生该事件。

实例代码如下：

```
<html>
```

```
<head>
<title>改变事件 onChange</title>
</head>
<body>
使用 onChange 设置改变状态的事件处理程序<br>
<form>
    <p>留言：<br></p>
    <textarea name="word" rows=5 cols=70 value=" " onchange=alert("您在文本框中添加了新的内容")>
    </textarea>
</form>
</body>
</html>
```

运行程序后，在文本框中添加任意文字后单击其他位置，会弹出添加文字的警告提示窗口，效果如图 12-17 所示。

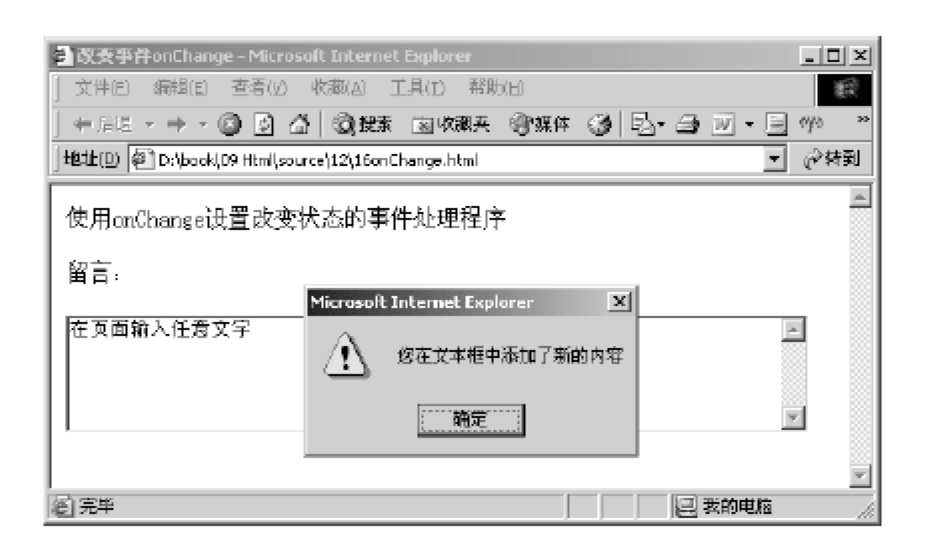


图 12-17　改变事件 onChange 的效果

12.4.4　选中事件——onSelect

当 text 或 textarea 对象中的文字被选中时会引发该事件。

实例代码：

```
<html>
<head>
<title>选中事件 onSelect </title>
</head>
<body>
使用 onSelect 设置选中文本的事件处理程序<br>
<form>
    <Input type="text" Value="文字信息" onSelect=alert("您选
中了文本框中的文字")>
</form>
</body>
</html>
```

运行程序，选中文本框中的部分或全部文字可以弹出提示选中了文本的警告提示窗口，如图 12-18 所示。

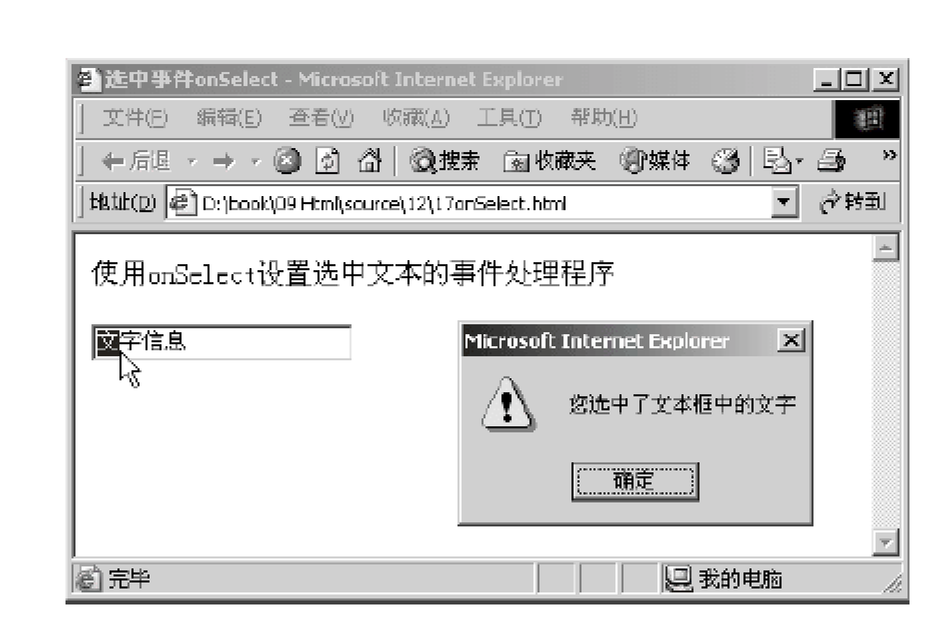


图 12-18　选中事件 onSelect 的效果

12.4.5　获得焦点事件——onFocus

当用户单击 text 或 textarea 以及 select 对象时，即将光标落在文本框或选择框时会产生该事件。

实例代码：

```
<html>
<head>
<title>焦点事件 onFocus</title>
<Script Language="JavaScript">
<!--
function befocus(){
alert("我被激活，成为了输入焦点");
}
-->
</Script>
</head>
<body>
使用 onFocus 设置输入焦点的事件处理程序<br><br>
选择一种证件名称：
<form>
    <select name="cardtype" onFocus="befocus()">
      <option value="id_card" selected>身份证
      <option value="stu_card">学生证
      <option value="drive_card">驾驶证
      <option value="other_card">其他证件
    </select>
</form>
</body>
</html>
```

运行程序后，单击下拉列表框时会弹出一个警告提示窗口，效果如图 12-19 所示。

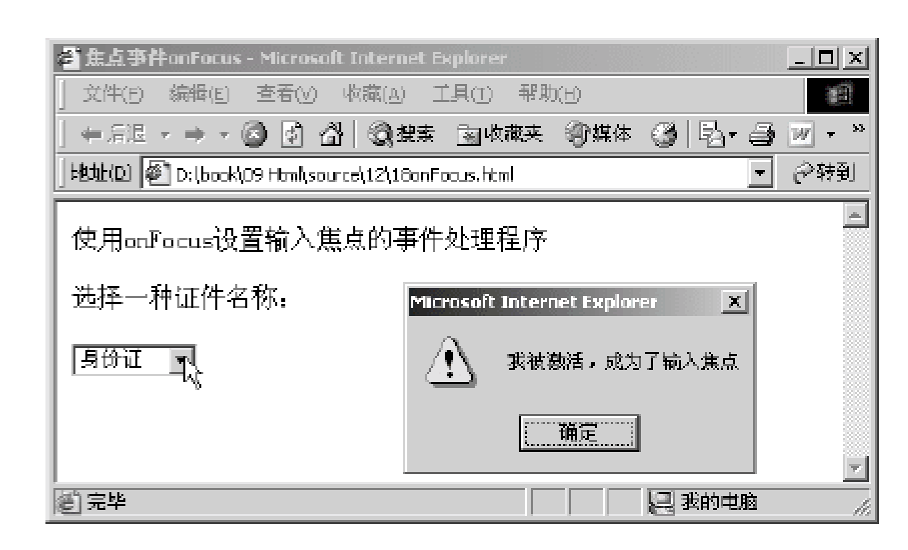


图 12-19　获得焦点事件 onFocus 的效果

12.4.6　失去焦点事件——onBlur

失去焦点事件正好与获得焦点事件相对，当 text 对象、textarea 对象或 select 对象不再拥有焦点而退到后台时，引发该事件。

实例代码：

```
<html>
<head>
<title>失去焦点事件 onBlur</title>
</head>
<body>
使用 onBlur 设置失去焦点事件的处理程序<br><br>
<form>
        <p>用户名：
          <input name="username" type="text" size=20 onBlur=confirm("确定“用户名”文本框失去焦点？")>
        </p>
        <p>登录密码：
          <input name="password" type="password" size=20 onBlur= confirm("确定“登录密码”文本框失去焦点？")>
        </p>
</form>
</body>
</html>
```

运行程序，将光标移动到任意一个文本框内，然后再将光标移至其他位置，此时会弹出确认窗口。如图 12-20 所示为光标离开“登录密码”文本框后的效果。



图 12-20　失去焦点 onBlur 的效果

12.4.7　载入文件事件——onLoad

当页面文件载入时，产生该事件。onLoad 事件的一个作用就是在首次载入一个页面文件时检测 cookie 的值，并用一个变量为其赋值，使它可以被源代码使用。本事件是 window 的事件，但是在 HTML 中指定事件处理程序时，一般把它写在<body>标记中。

实例代码：

```
<html>
<head>
<title>载入文件事件 onLoad</title>
</head>
<body onLoad=alert("正在载入页面，请耐心等待…")>
</body>
</html>
```

运行程序，效果如图 12-21 所示。

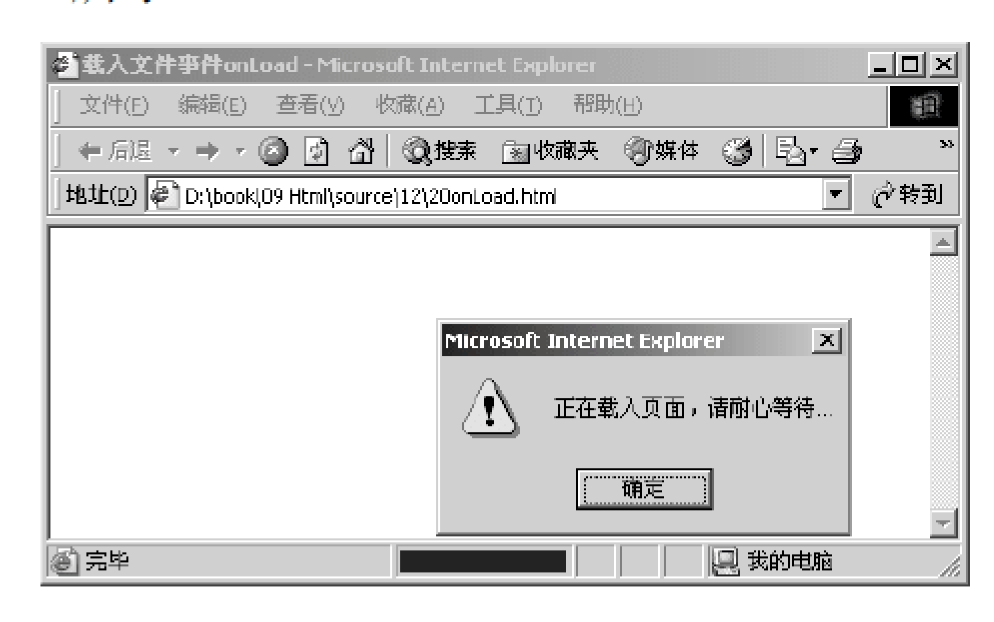


图 12-21　载入页面的效果

12.4.8　卸载文件事件——onUnload

卸载文件事件 onUnload 与载入文件事件正好相反，它是当 Web 页面退出时引发的事件，并可更新 cookie 的状态。

实例代码：

```
<html>
<head>
<title>卸载文件事件 onUnload</title>
</head>
<body onUnload=confirm("您确定要离开本页面么？")>
</body>
</html>
```

运行程序后，当要关闭该页面时会弹出提示对话框，效果如图 12-22 所示。



图 12-22　离开页面的效果

12.4.9　鼠标覆盖事件——onMouseOver

鼠标覆盖事件 onMouseOver 是当鼠标位于元素（如按钮）上方时所引发的事件。

实例代码：

```
<html>
<head>
<title>鼠标覆盖事件 onMouseOver</title>
</head>
<body>
将鼠标放在下面的按钮上可以改变浏览器的状态栏：<br><br>
     <form>
        <input type="button" value="改变页面状态栏"
         onmouseover="window.status='快来看看我是不是变了^_^';return true">
     </form>
</body>
</html>
```

运行程序，效果如图 12-23 所示，此时状态栏的文字为“完毕”；当鼠标位于按钮上方时，状态栏上的文字发生了变化，效果如图 12-24 所示。

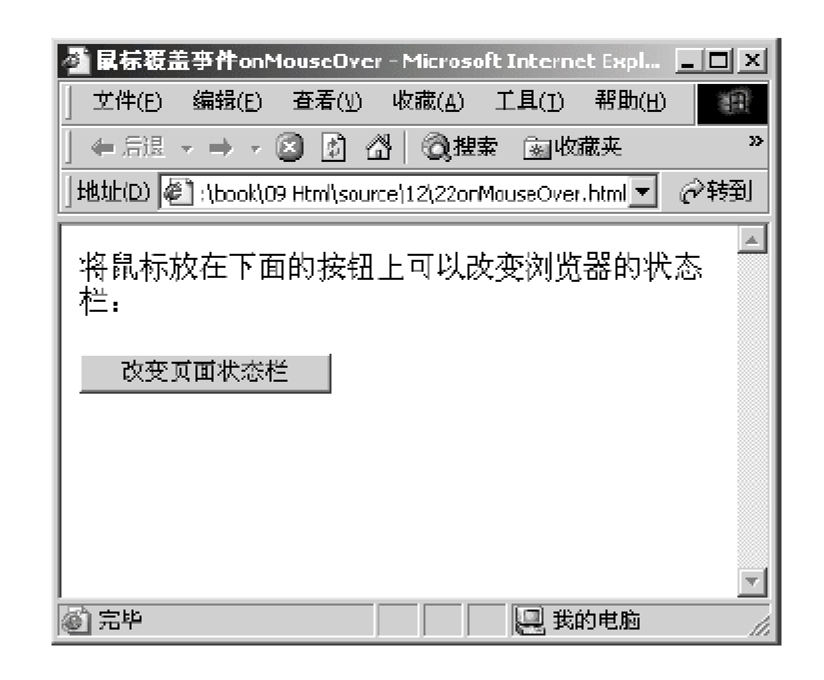


图 12-23　运行程序的效果

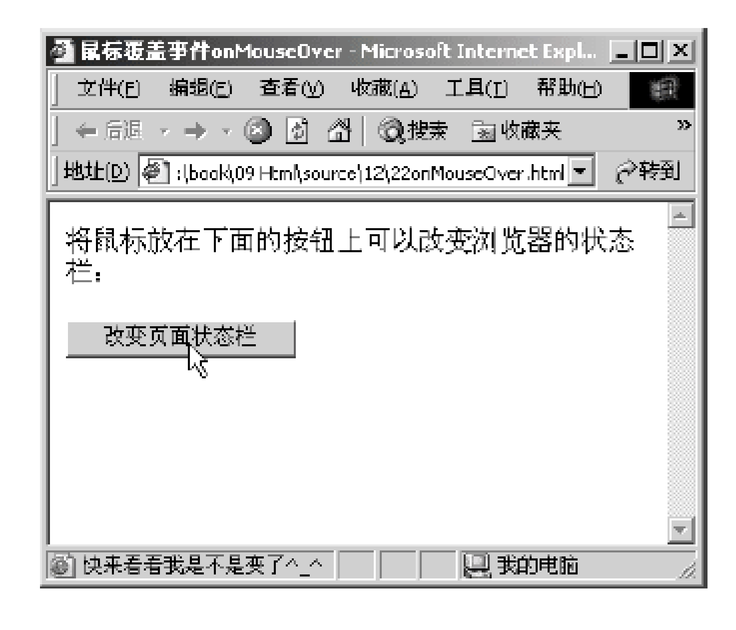


图 12-24　状态栏发生了变化

12.4.10　鼠标离开事件——onMouseOut

鼠标离开事件 onMouseOut 是当鼠标离开元素时引发的事件。如果它和鼠标覆盖事件同时使用，可以创建动态按钮的效果。

实例代码：

```
<html>
<head>
<title>鼠标覆盖事件 onMouseOver</title>
<Script Language="JavaScript">
<!--
function a(){
     eval("image").src="button0.png";
     event.srcElement.src="button1.png"
  }
function b(){
     eval("image").src="button1.png";
     event.srcElement.src="button2.png"
  }
-->
</Script>
</head>
<body>
动态按钮的效果：
<img id="image" src="button0.png" onMouseOver="a()" onMouseOut="b()">
</body>
</html>
```

运行程序，打开页面的效果如图 12-25 所示；当鼠标覆盖图像上方和离开图像上方时，效果分别如图 12-26 和图 12-27 所示。



图 12-25　运行程序的效果



图 12-26　鼠标覆盖的效果



图 12-27　鼠标离开的效果

12.4.11　其他事件

JavaScript 中还提供了一些其他事件，这里不再一一列举。下面通过表 12-4 对不同功能的事件进行简单描述。

表12-4　JavaScript中的一些常用事件

一 般 事 件	
事　件	**描　述**
onDblClick	鼠标双击事件
onMouseDown	鼠标上的按键被按下时激活的事件
onMouseUp	鼠标按下后，松开鼠标时触发的事件
onMouseMove	鼠标移动时触发的事件
onKeyPress	当键盘上的某个键被按下并且释放时触发的事件，要求页面内必须有激活的对象
onKeyDown	当键盘上某个按键被按下时触发的事件，要求页面内必须有激活的对象
onKeyUp	当键盘上某个按键被放开时触发的事件，要求页面内必须有激活的对象
页面相关事件	
事　件	**描　述**
onAbort	图片在下载时被用户中断
onBeforeUnload	当前页面的内容将要被改变时触发的事件
onError	捕捉当前页面因为某种原因而出现的错误，如脚本错误
onMove	浏览器的窗口被移动时触发的事件
onResize	当浏览器的窗口大小被改变时触发的事件
onScroll	浏览器的滚动条位置发生变化时触发的事件
onStop	浏览器的“停止”按钮被按下或者正在下载的文件被中断时触发的事件
表单相关事件	
事　件	**描　述**
onReset	当表单中reset属性被激活时触发的事件
onSubmit	一个表单被递交时触发的事件
滚动字幕事件	
事　件	**描　述**
onBounce	当Marquee内的内容移动至Marquee显示范围之外时触发的事件

续表

滚动字幕事件	
事　件	描　述
onFinish	当Marquee元素完成需要显示的内容后触发的事件
onStart	当Marquee元素开始显示内容时触发的事件
编 辑 事 件	
事　件	描　述
onBeforeCopy	当页面当前的被选择内容将要复制到浏览者系统的剪贴板前触发的事件
onBeforeUpdate	当浏览者粘贴系统剪贴板中的内容时通知目标对象
onContextMenu	当按下鼠标右键出现菜单时或者通过键盘的按键触发页面菜单时触发的事件，例如在页面的<body>中加入onContentMenu="return false"可禁止使用鼠标右键
onCopy	当页面当前被选择内容被复制后触发的事件
onCut	当页面当前被选择内容被剪切时触发的事件
onPaste	当内容被粘贴时触发的事件
onDrag	当某个对象被拖动时触发的事件
onDragEnd	当鼠标拖动结束时触发的事件，即鼠标的按键被释放时触发的事件
数 据 绑 定	
事　件	描　述
onAfterUpdate	当数据完成由数据源到对象的传送时触发的事件
onCellChange	当数据来源发生变化时触发的事件
onDataAvailable	当数据接收完成时触发的事件
onDatasetChanged	数据在数据源发生变化时触发的事件
onDatasetComplete	当数据源的全部有效数据读取完毕时触发的事件
onErrorUpdate	当使用onBeforeUpdate事件触发取消了数据传送时，代替onAfterUpdate事件
onRowEnter	当前数据源的数据发生变化并且有新的有效数据时触发的事件
onRowExit	当前数据源的数据将要发生变化时触发的事件
onRowsDelete	当前数据记录将被删除时触发的事件
onRowsInserted	当前数据源将要插入新数据记录时触发的事件
外 部 事 件	
事　件	描　述
onAfterPrint	当文档被打印后触发的事件
onBeforePrint	当文档即将打印时触发的事件
onHelp	当浏览者按下F1键或者单击浏览器中的“帮助”按钮时触发的事件

12.5　利用 JavaScript 制作特效

实际上，在创建网页时常常利用 JavaScript 制作一些特效，从而为页面增添活力。本节将介绍一些利用 JavaScript 程序编写的常用特效，这些特效可以直接粘贴到网页代码的头部或主体内，可以根据需要对代码进行适当的修改。

提示：本书只给出 JavaScript 的特效代码，而网页内容的代码不影响其效果，这里不提供网页内容的代码，读者可以直接将代码复制到自己的页面中加以利用。对于特效中的文字、颜色等，可以根据需要进行修改。

12.5.1　动态显示时间

在很多网页上都能显示当前的时间，这主要是通过调用浏览器中附带的计时器（setTimeout）来实现的。如果设定了每一段时间更新一次网页，就可以同时刷新页面的时间。实例代码如下：

```
<html>
<head>
<title>在页面的状态栏动态显示时间</title>
</head>
<body onLoad="showtime();" onunload="clearTimeout(tID);">
在状态栏上显示了当前时间
<Script Language="JavaScript">
var gtime;
function showtime() {
    var now = new Date();
<--取得当前小时、分、秒-->
    var hours = now.getHours();
    var minutes = now.getMinutes();
    var seconds = now.getSeconds();
    var timeValue = "现在是"+((hours >= 12) ? "下午" : "上午" );
    timeValue +=((hours >12) ? hours -12 :hours);
    timeValue += "点  " + minutes + "分  " + seconds + "秒";
    window.status = timeValue;
    <--每 1000 毫秒更新一次-->
    gtime = setTimeout("showtime()",1000);
}
</Script>
</body>
</html>
```

运行程序，可以看到如图 12-28 所示的效果。

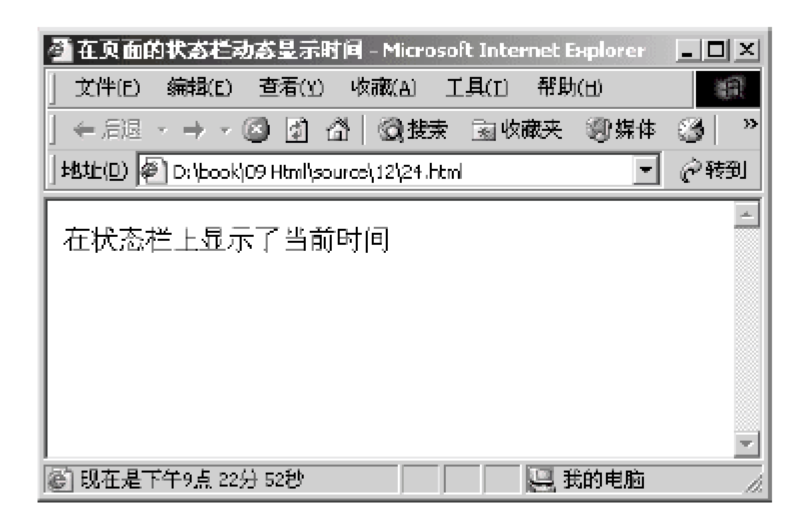


图 12-28　在状态栏显示当前时间

12.5.2　随鼠标移动的图像

随鼠标移动的图像是一种特殊的动态鼠标效果，一般会有一些动态图像跟随鼠标运动。读者可以直接调用 JavaScript 的代码来设置鼠标效果，具体代码如下：

```
<html>
<head>
<title>跟随鼠标的图像</title>
</head>
<body>
<Script Language="JavaScript">
```

```
    var newtop=0         <--新的上边缘坐标-->
    var newleft=0        <--新的左边缘坐标-->
    layerStyleRef="layer.style.";
    layerRef="document.all";
    styleSwitch=".style";
    function doMouseMove()  <--响应鼠标的移动-->
    {
        layerName = 'picture'
        eval('var curElement='+layerRef+'[" '+layerName+' "]')
        eval(layerRef+'[" '+layerName+' "]'+styleSwitch+'.visibility="hidden" ')
        eval('curElement'+styleSwitch+'.visibility="visible" ')
        eval('newleft=document.body.clientWidth-curElement'+styleSwitch+'.pixelWidth')
        eval('newtop=document.body.clientHeight-curElement'+styleSwitch+'.pixelHeight')
        eval('height=curElement'+styleSwitch+'.height')
        eval('width=curElement'+styleSwitch+'.width')
        width=parseInt(width)
        height=parseInt(height)

        if (event.clientX > (document.body.clientWidth - width))
        {
        newleft=document.body.clientWidth + document.body.scrollLeft- width
        }
        else{newleft=document.body.scrollLeft + event.clientX}

        eval('curElement'+styleSwitch+'.pixelLeft=newleft')

        if (event.clientY > (document.body.clientHeight - height))
        {
        newtop=document.body.clientHeight + document.body.scrollTop - height
        }
        else
        {
        newtop=document.body.scrollTop + event.clientY
        }
        eval('curElement'+styleSwitch+'.pixelTop=newtop')
    }
    document.onmousemove = doMouseMove;
    document.write("<img ID=picture src='pic01.gif' Style='position:absolute;TOP:0pt;LEFT:0pt;
                    Z-INDEX:2; visibility:hidden;'>")
</Script>
</body>
</html>
```

运行程序，效果如图 12-29 所示。



图 12-29　随鼠标运动的图像

12.5.3　禁止鼠标右击

在一些网页上，当用户单击鼠标右键时会弹出警告窗口或者直接没有任何反应。这种功能可以使用如下代码实现：

```
<html>
<head>
<title>禁止鼠标右键单击</title>
<Script Language="javascript">
<--设置警告窗口-->
function click() {
if (event.button==2) {
alert('警告窗口的内容')
<--如果 alert 的内容为空，那么在右击时就会没有任何反应-->
}
}
<--调用定义的函数-->
document.onmousedown=click
</Script>
</head>
<body>
本页面禁止使用鼠标右键
</body>
</html>
```

运行程序，效果如图 12-30 所示。



图 12-30　禁止鼠标右键

12.5.4　设置链接文字的自动变色

如果一个页面的链接文字不停地变换颜色，那么该链接一定会吸引到用户的注意。实现方法也很简单，代码如下：

```
<html>
<head>
<title>设置链接文字的自动变色</title>
</head>
```

```
<body bgcolor="#DDCCDD">
<Script Language="JavaScript">
    <--/判断颜色的个数-->
    function initArray() {
      for (var i = 0; i < initArray.arguments.length; i++) {
        this[i] = initArray.arguments[i];
      }
    this.length = initArray.arguments.length;
    }
    var colors = new initArray("coral","darkcyan",
                "darkorchid","black","orange","darkgreen","cornsilk","coral");
    <--设置变化的时间-->
    delay = 0.5;
    <--设置链接文字的变色效果-->
    link = 0;
    vlink = 0;
    function linkDance() {
      link = (link+1)%colors.length;
      vlink = (vlink+1)%colors.length;
      document.linkColor = colors[link];
      document.vlinkColor = colors[vlink];
      setTimeout("linkDance()",delay*1000);
    }
    linkDance();
</Script>
    <p>查看链接文字的颜色：
    <p><a href="exam.html">会变色的链接文字</a>
</body>
</html>
```

运行程序，效果如图 12-31 所示。读者在使用时只需要将<script>标记和</script>标记内的代码复制到页面的主体部分（即<body>和</body>之间）即可。而不同的颜色设置可以产生不同的变色效果。



图 12-31　自动变色的链接文字

12.5.5　显示浏览器信息

本实例使用 JavaScript 制作了一个显示浏览器信息的窗口，当打开页面之后，在页面顶部的表格内就会出现关于用户浏览器的详细信息。下面就来看一下这个实例的具体实现步骤。

```
……
<script>
function whatBrowser()
{
        <!--检测浏览器的名称-->
        document.Browser.Name.value=navigator.appName;
        <!--检测浏览器的版本号-->
        document.Browser.Version.value=navigator.appVersion;
        <!--检测浏览器的代码名称-->
        document.Browser.Code.value=navigator.appCodeName;
        <!--检测浏览器的用户代理标识-->
        document.Browser.Agent.value=navigator.userAgent;
}
</script>
<table border >
<form name ="Browser">
<tr>
<td>浏览器名称:</td>                         <!--在表格中输出浏览器的名称-->
<td><input type ="txt" name="Name" size="45"></td>
</tr>
<tr>
<td>版本号:</td>                             <!--在表格中输出浏览器的版本号-->
<td><input type="txt" name="Version" size="45"></td>
</tr>
<tr>
<td>代码名称:</td>                           <!--在表格中输出浏览器的代码名称-->
<td><input type="txt" name="Code" size="45"></td>
</tr>
<tr>
<td>用户代理标识:</td>                       <!--在表格中输出浏览器的用户代理标识-->
<td><inpet type ="txt" name="Agent" size="45"></td>
</tr>
</form>
</table>
<body onLoad="whatBrowser()">                <!--调用 what B rowser()函数-->
……
```

本程序的难点主要是实现检测浏览器的详细信息，并将其显示在页面的表格内。显示的结果如图 12-32 所示。



图 12-32　显示浏览器的详细信息

12.5.6　显示访客登录信息的窗口

本实例使用 JavaScript 制作了一个显示访客登录信息的窗口，在用户提示框内输入用户名后页面就会显示上次登录的详细信息。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
var expDays = 10;                                                  <!--cookie 的期限-->
var exp = new Date();                                              <!--新的时间变量-->
exp.setTime(exp.getTime() + (expDays*24*60*60*1000));              <!--过期的日期-->
function Who(info)                                                 <!--获得访问者的昵称-->
{
        <!--先从 cookie 中查找 visitorName 字段-->
        var VisitorName = GetCookie('VisitorName')
        if (VisitorName == null)                     <!--如果没有找到，则提示用户输入昵称-->
        {
          VisitorName = prompt("请输入你的昵称：",",");                  <!--弹出提示窗口-->
          <!--将用户输入的昵称写入 cookie 中-->
          SetCookie ('VisitorName', VisitorName, exp);
        }
        return VisitorName;                                        <!--返回用户的昵称-->
}
function When(info)                                                <!--返回用户的上次访问时间-->
{
        var rightNow = new Date()                                  <!--构造变量-->
        var WWHTime = 0;
        WWHTime = GetCookie('WWhenH')                 <!--从 cookie 中获取上一次的时间-->
        WWHTime = WWHTime * 1
        var lastHereFormatting = new Date(WWHTime);
        <!--转化时间格式-->
        var intLastVisit = (lastHereFormatting.getYear() * 10000)+(lastHereFormatting.getMonth() * 100) +
lastHereFormatting.getDate()
        var lastHereInDateFormat = "" + lastHereFormatting;
        var dayOfWeek = lastHereInDateFormat.substring(0,3)          <!--获取日子-->
        var dateMonth = lastHereInDateFormat.substring(4,11)         <!--获取日期-->
        var timeOfDay = lastHereInDateFormat.substring(11,16)        <!--获取时间-->
        var year = lastHereInDateFormat.substring(23,25)             <!--获取年份-->
        <!--按日、月、年重新组合日期-->
        var WWHText = dayOfWeek + " " + dateMonth+ timeOfDay
        <!--将用户访问时间和过期时间均写入 cookie 中-->
        SetCookie ("WWhenH", rightNow.getTime(), exp)
        return WWHText;
}
function Count(info)                                               <!--返回用户上站次数-->
{
        var WWHCount = GetCookie('WWHCount')     <!--从 cookie 中读取 wwwhcount 字段-->
        if (WWHCount == null)                                      <!--如果返回值为空-->
        {
          WWHCount = 0;                                            <!--上站次数为 0-->
        }
```

```
        else
        {
          WWHCount++;                                          <!--否则，上站次数加 1-->
        }
        <!--同时将更新过的信息写入 cookie 中-->
        SetCookie ('WWHCount', WWHCount, exp);
        return WWHCount;                                       <!--返回上站次数-->
}
function set()
{
        VisitorName = prompt("请输入你的新昵称：",",");       <!--弹出窗口，提示输入昵称-->
        <!--将用户昵称写入 cookie 中-->
        SetCookie ('VisitorName', VisitorName, exp);
        SetCookie ('WWHCount', 0, exp);                        <!--将用户上站次数 0 写入 cookie 中-->
        SetCookie ('WWhenH', 0, exp);                          <!--将用户最后上站时间写入 cookie 中-->
}
function getCookieVal (offset)                                 <!--返回 cookie 中一定位置的字段-->
{
        <!--获取指定位置字段的末尾的位置-->
        var endstr = document.cookie.indexOf (";", offset);
        if (endstr == -1)                                      <!--如果返回值为－1-->
        endstr = document.cookie.length;                       <!--则将给定位置后的所有字符串均返回-->
        <!--返回字符串-->
        return unescape(document.cookie.substring(offset, endstr));
}
function GetCookie (name)                                      <!--从 cookie 中获取字段的值-->
{
        var arg = name + "=";                                  <!--变量-->
        var alen = arg.length;                                 <!--字段的长度-->
        var clen = document.cookie.length;                     <!--cookie 的长度-->
        var i = 0;
        while (i < clen)                                       <!--在 cookie 中-->
        {
          var j = i + alen;                                    <!--移动一个字段的宽度-->
          if (document.cookie.substring(i, j) == arg)          <!--如果是待找字段-->
          return getCookieVal (j);                             <!--返回字段的值-->
          i = document.cookie.indexOf(" ", i) + 1;             <!--更新 i 的值-->
          if (i == 0)
          break;                                               <!--如果找完了，返回-->
        }
        return null;
}
function SetCookie (name, value)                               <!--向 cookie 中写入-->
{
        var argv = SetCookie.arguments;                        <!--写入的变量数组-->
        var argc = SetCookie.arguments.length;                 <!--变量的个数-->
        <!--变量数组中的第二个表示过期的时间-->
        var expires = (argc > 2) ? argv[2] : null;
        var path = (argc > 3) ? argv[3] : null;                <!--变量的第三个表示路径-->
        var domain = (argc > 4) ? argv[4] : null;              <!--变量的第四个表示域-->
        var secure = (argc > 5) ? argv[5] : false;             <!--变量的第五个表示权限-->
        document.cookie = name + "=" + escape (value) + ((expires == null) ? "" : ("; expires="
```

```
+expires.toGMTString())) + ((path == null) ? " " : ("; path=" + path)) + ((domain == null) ? " " : ("; domain=" +
domain)) + ((secure == true) ? "; secure" : " ");                              <!--按照给定的格式写入 cookie-->
}
</Script>
</head>
<body>
<Script Language="JavaScript">
document.write("您好 <b>" + Who() + "</b>，您是第 <b>" + Count() + "</b> 次访问本主页. <br>最后一次是
<b>" + When() +"</b>")
document.write('<a href="JavaScript:set()">[按此改名]</a>')
</Script>
</body>
<body>
</body>
</html>
……
```

本程序实现了显示访客登录信息的功能，访客的登录信息包括访客的昵称、访问的次数以及上次访问的时间，难点主要是时间函数以及字符串函数的使用方法。显示的结果如图 12-33 和图 12-34 所示。



图 12-33　用户提示框



图 12-34　显示用户详细信息

12.5.7　标题渐变的窗口

本实例使用 JavaScript 制作了一个标题渐变的窗口效果，当页面被打开后标题就会按由大到小再到大的过程变化。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
var speed = 2;                                  <!--速度-->
var cycledelay = 1000;                          <!--循环周期-->
var maxsize = 48;                               <!--最大数目-->
var x = 0;
var y = 0;
var themessage, size;
var esize = " ";
function initArray()                            <!--初始化数组-->
{
  this.length = initArray.arguments.length;
  for (var i = 0; i < this.length; i++)
  {
```

```
        this[i] = initArray.arguments[i];
    }
}
var themessage2 = new initArray(                       <!--初始化矩阵-->
"欢迎光临",
"JavaScript 世界",
"隆重奉献",
"网页特效",
"色彩缤纷",
"精彩呈现"
);
if (navigator.appVersion.indexOf("MSIE") != -1)
    document.write('<span id="wds"></span><br>');      <!--新的层-->
function upwords()
{
    themessage = themessage2[y];                       <!--读取第 y 条信息-->
    if (x < maxsize)                                   <!--当前字体大小小于最大值-->
    {
        x++;                                           <!--字体变大-->
        setTimeout("upwords()",speed);
}
    else                                               <!--当前字体已经最大-->
    setTimeout("downwords()",cycledelay);              <!--字体变小-->
    if (navigator.appVersion.indexOf("MSIE") != -1)
    {
        wds.innerHTML = "<center>"+themessage+"</center>";   <!--写入信息-->
        wds.style.fontSize=x+'px'                      <!--按照给定的字体-->
  }
}
function downwords()
{
    if (x > 1)                                         <!--如果当前字体比 1 大-->
    {
      x--;                                             <!--字体继续变小-->
      setTimeout("downwords()",speed);
}
else                                                   <!--字体已经最小-->
{
    setTimeout("upwords()",cycledelay);                <!--更换下一条信息-->
      y++;                                             <!--下一条信息-->
      if (y > themessage2.length - 1) y = 0;           <!--如果到头，重头开始-->
}
if (navigator.appVersion.indexOf("MSIE") != -1)
{
      wds.innerHTML = "<center>"+themessage+"</center>";     <!--写入信息-->
      wds.style.fontSize=x+'px'                        <!--按照给定的字体-->
```

```
    }
  }
  setTimeout("upwords()",speed);                    <!--调用 upwords()函数-->
  </Script>
  ……
```

本程序实现了标题渐变的窗口效果，难点主要是字体在窗口中变化的效果的实现以及循环处理、显示消息的方法。显示的结果如图 12-35 和图 12-36 所示。



图 12-35　标题渐变的窗口（1）



图 12-36　标题渐变的窗口（2）

12.5.8　网页中的 loading 条

本实例使用 JavaScript 制作了一个网页中的 loading 条，该 loading 条在页面打开之后就开始读取，在到达 100%之后会转到指定的页面。下面就来看一下这个实例的具体实现步骤。

```
……
    <form name=loading>
    <p align=center> <font color="#800080" size="2">载入中，请稍等</font><font color="#FFFF00" size="2"
face="Arial">...</font>
<input type=text name=chart size=46 style="font-family:Arial; font-weight:bolder; color:#800080;
background-color:#C0FFFF; padding:0px; border-style:none;">
    <input type=text name=percent size=47 style="color:#800080; text-align:center; border-width:medium;
border-style:none;">
    <Script Language="JavaScript">
    var bar=0                                        <!--声明变量-->
    var line="||"                                    <!--loading 条的样式-->
    var amount="||"                                  <!--loading 条的样式-->
    count()                                          <!--直接调用 count()函数-->
function count()
{
    bar=bar+2                                        <!--每次增加两个 bar-->
    amount =amount + line                            <!--loading 条每次增加两条竖线-->
    document.loading.chart.value=amount              <!--刷新进展条-->
    document.loading.percent.value=bar+"%"           <!--计算出百分比-->
    if (bar<99)
    {
      setTimeout("count()",100);                     <!--每 0.1 秒进展条加 1-->
```

```
        }
        else
        {
            <!--进展条满后，显示另一个网页-->
            window.location = "http://www.JavaScript.com/";
        }
    }
</Script>
……
```

本程序实现了进展条的功能，难点主要是进展条的具体实现过程以及相应的进展百分比的实现过程，可以根据不同的情况输出不同的网页。显示的结果如图 12-37 和图 12-38 所示。

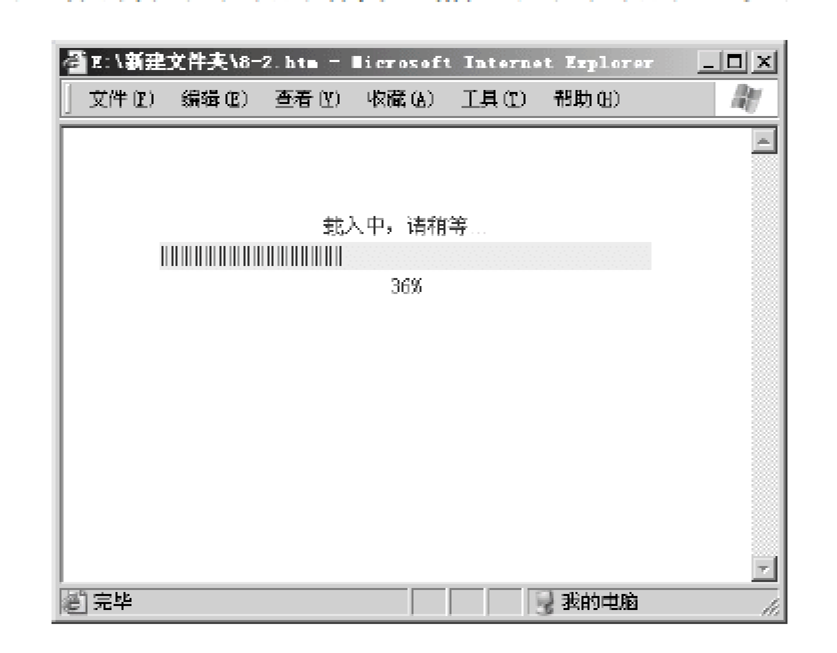


图 12-37　网页中的 loading 条



图 12-38　打开指定的页面

12.5.9　页面的制作完成时间

本实例使用 JavaScript 制作了一个显示页面制作完成时间的页面，该页面可以根据页面的初始完成日期来计算到现在为止所完成的时间。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language ="JavaScript">
function HowLongSince(startmonth, startdate, startyear)
{
    sdate=startdate;                                    <!--获得开始的日期-->
    smonth=startmonth-1;                                <!--获得开始的月份-->
    syear=startyear;                                    <!--获得开始的年份-->
    <!--每个月中包含的天数-->
    var DaysInMonth = new Array(31,28,31,30,31,30,31,31,30,31,30,31);
    today = new Date()                                  <!--获得当前的日期-->
    var thisyear = today.getFullYear();                 <!--获得当前的年份-->
    var thismonth = today.getMonth();                   <!--获得当前的月份-->
    var thisdate = today.getDate();                     <!--获得当前的日期-->
    mstart = new Date(syear,(smonth==12?1:smonth+1),1); <!--新的 Date 变量-->
    <!--获得经历的天数 1-->
    days1 = (mstart - new Date(syear,smonth,sdate))/(24*60*60*1000)-1;
    mend = new Date(thisyear,thismonth,1);
    <!--获得经历的天数 2-->
    days2 = (new Date(thisyear,thismonth,thisdate) - mend)/(24*60*60*1000)+1;
```

```
    dayst = days1 + days2;                      <!--经历的天数等于前两个之和-->
  if (dayst >= DaysInMonth[smonth])
  {
      <!--如果经历的天数大于该月的天数，则增加一个月，同时天数减少一个月的天数-->
      AddOneMonth = 1;
      dayst -= DaysInMonth[smonth];
   }
   else AddOneMonth = 0;                                 <!--否则不增加-->
   ydiff1 = thisyear-mstart.getFullYear();               <!--获得经历的年数-->
   mdiff1 = thismonth-mstart.getMonth()+AddOneMonth;     <!--获得经历的月份数-->
   if (mdiff1 >11)
   {
     mdiff1=0; ydiff1++;                                 <!--如果月份超过 12，则加一年-->
   }
   if (mdiff1 < 0)
   {
     mdiff1 = mdiff1 + 12; ydiff1--;                     <!--如果月份小于 0，则减一年-->
   }
   <!--返回当前年数-->
   temp = (ydiff1==0?"":(ydiff1==1?ydiff1+"年零":ydiff1 + "年零"));
   <!--返回当前月数-->
   temp += (mdiff1==0?"个月又 ":(mdiff1==1?mdiff1+"个月又":mdiff1+"个月又"));
   <!--返回当前天数-->
   temp += (dayst==0?"天！ ":(dayst==1 ? " 天！ " : dayst + "天！ " ));
   return temp;
  }
  document.write("本页面已经制作完成:");
  document.write("<font color=red>");                    <!--设置字体颜色-->
  document.write(HowLongSince(11,10,2002));              <!--调用函数 HowLongSince()-->
  document.write("</font>");
  </Script>
  ……
```

本程序实现了页面制作完成时间的功能，难点主要是能够将时间转换成年、月、日的形式。显示的结果如图 12-39 所示。



图 12-39　带有农历的日历

12.5.10　在状态栏显示输入字符的页面

本实例使用 JavaScript 制作了一个在状态栏显示输入字符的页面，在页面打开之后，会弹出一个用户提示框，在提示框内输入一些内容，然后就会显示在状态栏内。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
<!--弹出窗口，用户可以输入字符-->
Ret = prompt('输入你想在状态栏上显示的字符',"^_^")
var temp
var f =" "                                    <!--定义一段空格的字符串-->
var f = f + (Ret)                             <!--将输入的字符加在空格字符串后边-->
var speedtogo = 50                            <!--定义刷新速率-->
var counter
function scrollon()
{
   temp   = f.substring(0,1);                 <!--获取第一个字符-->
   f += temp                                  <!--将第一个字符放在字符串的最后-->
   f= f.substring(1,100);                     <!--输出后 100 个字符-->
   window.status = f.substring(0,100);        <!--在状态栏中显示输入的内容-->
   counter = setTimeout("scrollon()",speedtogo);   <!--按照设置的时间延时-->
}
</Script>
<body onLoad="scrollon()">
……
```

本程序实现了输入字符出现在状态栏的功能，难点主要是利用弹出窗口获取用户输入并在状态栏中显示字符。显示的结果如图 12-40 和图 12-41 所示。

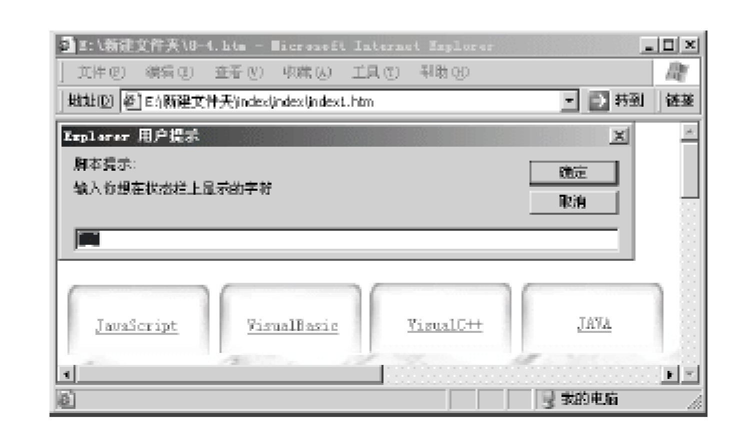


图 12-40　状态栏显示输入的字符（1）



图 12-41　状态栏显示输入的字符（2）

12.5.11　页面的加密功能

本实例使用 JavaScript 制作了一个页面的加密功能，如果需要查看网页内容，则必须输入正确的密码才可以通过。下面就来看一下这个实例的具体实现步骤。

```
……
  <Script Language="JavaScript">
  al="`1234567890-=~!@#$%^&*()_+qwer"+"tyuiop[]QWERTYUIOP{}|asdfghjkl;A"+"SDFGHJKL:zxcvbnm,
```

```
  ./ZXCVBNM<>?";
ab1=" ";
bctr=0;
function ckPwd()
{
   <!--获得输入的用户名和密码-->
   tst=document.isn.username.value +"*"+document.isn.passwrd.value+"*";
   ls=document.pd.pe.value;                              <!--获得 pe 的值-->
   a=eval(ls.substring(0,2))-91;
   ls=ls.substring(2,ls.length);                         <!--去掉前两位-->
   nls=" ";                                              <!--nls 变量为空-->
   flg=0;                                                <!--flg 标志为 0-->
   while (ls.length>12)                                  <!--如果 ls 的长度大于 12-->
   {
      ab=eval(ls.substring(0,2))-89;                     <!--ls 前两位的值减去 89 后赋给 ab-->
      ab1=(ab1==" "?" "+ab:ab1);                         <!--如果 ab1 为空,则把 ab 的值赋给 ab1-->
      oab1=ab1;                                          <!--把 ab1 的值赋给 oab1-->
      ls=ls.substring(2,ls.length);                      <!--去掉前面两位-->
      for (var i=0;i<ab;i++)                             <!--循环 ab 次-->
      {
          nr=eval(ls.substring(0,2))-a;                  <!--将 ls 前两位的值减去 a 的值赋给 nr-->
          ls=ls.substring(2,ls.length);                  <!--去掉 ls 的前两位-->
          nls+=al.charAt(nr);                            <!--nls 自身加上 al 中出现 nr 字符的位置-->
       }
       nls+="*";                                         <!--nls 中加入 ‘*’ -->
       if (nls.indexOf(tst)>-1)                          <!--如果 nls 中出现了 tst 的字样-->
       {
          ls=" ";                                        <!--ls 清空-->
          flg=1;                                         <!--flg 标志位变成 1-->
          }
}
if (flg==1)
{
  tstOk();                                        <!--密码通过-->
}
else
{
   bctr++;                                        <!--否则，验证的次数加 1-->
   if (bctr>3)                                    <!--如果验证次数超过 3-->
   {
     location.href="nopass.htm";                  <!--页面导入到失败的页面-->
   }
   else
   {
        <!--否则，弹出窗口提示用户-->
        alert("对不起,用户名/密码错误."    +" 你已试登录"+bctr+"次.");
   }
 }
```

```
}
function tstOk()
{
     ab1=ab1+" "+a;
     alert("正确，开始登录!");                          <!--弹出窗口，提示用户-->
     location.href="pwds.htm?"+ab1;                    <!--转入到指定页面-->
}
</Script>
<form name ="pd">
<input type='hidden' name='pe'
  value='9393693658789606190557772053938953313395586858685868976834793758313033975834793758313033'>
</form>
<form name ="isn">
<table border =2 cellpadding=5 cellspacing=0 bgcolor=#c0c0c0>
<tr><td colspan=2 aling=center><font size=4 color=white face="楷体_GB2312"><b>请您输入
</b></font></td></tr>
<tr><td><font size=3 color=white><b>用户名：</b></font></td>
<td><input type ="text" name="username" value=" " size=15></td></tr>
<tr><td><font size=3 color=white><b>密  码：</b></font></td>
<td><input type="password" name="passwrd" value=" " size=15></td></tr>
<tr><td colspan=2 aling=center><input type="button" name="btn" value =" 登录 " onClick="ckPwd();return
false;"></td></tr>
</table></form>
<table border=1 width=486 bgcolor=#c0c0c00>
 <tr><td colspan=3 alting= center >< font face="楷体_GB2312" color=white><b>你可以使用如下号码:</n>
</font></td</tr>
 <tr><td>< font face="helvetica,arial,geneva" color=white>用户名：John<br>密    码：
4$3gb%a</td>
 <td>< font face ="helvetica,arial,geneva" color = white >用户名：Mary<br>密    码：
hHhHhH</td>
 <td>< font face ="helvetica,arial,geneva" color = white >用户名：Humphrey<br>密    
码：humphrey</td></tr>
</table>
……
```

本程序的难点主要是实现了页面加密的功能，密码的验证过程是本例的重点，应掌握根据不同的情况，给出不同输出结果的方法。显示的结果如图 12-42 和图 12-43 所示。



图 12-42　页面的加密功能（1）



图 12-43　页面的加密功能（2）

12.5.12　调色板更换页面背景

本实例使用 JavaScript 制作了一个调色板用于更换页面背景，在页面打开之后，会出现 6 个调色板，单击某一种颜色，背景就会相应地显示出来。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
var hex = new Array(6)                                    <!--定义数组变量-->
hex[0] = "FF"
hex[1] = "CC"
hex[2] = "99"
hex[3] = "66"
hex[4] = "33"
hex[5] = "00"
function display(triplet)
{
  <!--根据输入的 triplet，更新窗口的背景颜色-->
  document.bgColor = '#' + triplet
  alert('现在的背景颜色代码是：  ' + triplet)              <!--弹出提示窗口-->
}
function drawCell(red, green, blue)
{
  <!--定义小方格内的背景颜色-->
  document.write('<TD BGCOLOR="#' + red + green + blue + '">')
  document.write('<AHREF="javascript:display(\''+(red+green+blue)+'\')">')
  <!--定义超链接，调用 display()函数-->
  <!--引用图片-->
  document.write('<IMG SRC="place.gif" BORDER=0 HEIGHT=12 WIDTH=12>')
  document.write('</A>')
  document.write('</TD>')
}
function drawRow(red, blue)
{
  document.write('<TR>')
  for (var i = 0; i < 6; ++i)                               <!--循环 6 次-->
  {
    <!--一行 6 个小方格内部的颜色差别在于 red 和 blue 分量-->
    drawCell(red, hex[i], blue)
  }
  document.write('</TR>')
}
function drawTable(blue)
{
  document.write('<TABLE CELLPADDING=0 CELLSPACING=0 BORDER=0>')
  for (var i = 0; i < 6; ++i)                               <!--循环 6 次-->
  {
    drawRow(hex[i], blue)           <!--每个表格总体的颜色差别在于蓝色分量的不同-->
```

```
    }
      document.write('</TABLE>')
  }
  function drawCube()
  {
     document.write('<TABLE CELLPADDING=5 CELLSPACING=0 BORDER=1><TR>')
     for (var i = 0; i < 6; ++i)
     {
        <!--定义表格的背景颜色为白色-->
        document.write('<TD BGCOLOR="#FFFFFF">')
        drawTable(hex[i])                              <!--循环画出每一个表格-->
        document.write('</TD>')
        }
        document.write('</TR></TABLE>')
}
  drawCube()                                   <!--直接调用 drawCube()函数-->
  </Script>
  <body>
  单击上面的调色板试试
  </body>
  </html>
  ……
```

本程序实现了使用调色板来更换背景的功能，难点主要是颜色的基本构成以及调配方法。显示的结果如图 12-44 和图 12-45 所示。

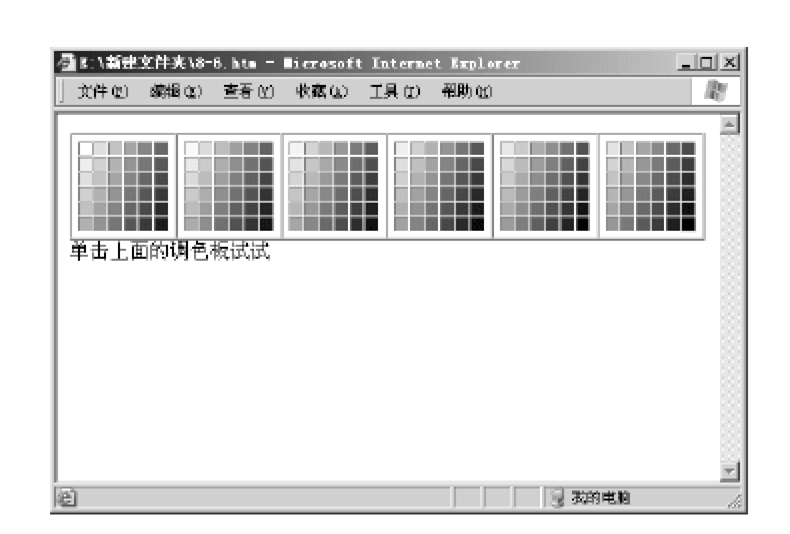


图 12-44　调色板来更换背景（1）



图 12-45　调色板来更换背景（2）

12.5.13　自由移动的图片

本实例使用 JavaScript 制作了一个可以自由移动的图片，当要移动图片时，在文本框内输入相应的位置，单击按钮即可移动到相应的位置。下面就来看一下这个实例的具体实现步骤。

```
……
<style type="text/css">
#moveobj
{
position: relative;
}
</style>
```

```
<Script Language="JavaScript">
function moveit()                                          <!--定义函数 moveit-->
{
  <!--设置图片距离顶端的位置-->
  moveTop = document.forms[0].elements[0].value;
  <!--设置图片距离左端的位置-->
  moveLeft = document.forms[0].elements[1].value;
  moveobj.style.top = moveTop;
  moveobj.style.left = moveLeft;
}
</Script>
<div id="moveobj"><img src="bike.jpg"></div><p>
<form action="javascript:moveit()">
顶边距离:<input type=text size=3 name=topnum value=0>        <!--设置文本框-->
左边距离:<input type=text size=3 name=leftnum value=0>       <!--设置文本框-->
<input type=submit value="移动" name="submit">               <!--设置按钮-->
</form>
……
```

本程序实现了图片的自由移动，难点主要是当在文本框内输入数值时，页面的图片就要移动到相应的位置。显示的结果如图 12-46 和图 12-47 所示。



图 12-46　自由移动的图片（1）



图 12-47　自由移动的图片（2）

12.5.14　图片代替按钮效果

本实例使用 JavaScript 制作了一个图片代替按钮的效果，在通常情况下，按钮都是一个带有字体标记的长方形，在这里将用图片来代替这些常规的按钮。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language=JavaScript>
<!-- Hide the script from old browsers --
function surfto(form)
{
  var myindex=document.myform.dest.selectedIndex
  <!--设置与图片的链接-->
  location=document.myform.dest.options[myindex].value;
}
</Script>
```

```
<form name="myform">
<select name="dest" size=1>
<option selected>选项</option>
<option value="http://www.sohu.com">搜狐网站</option>         <!--设置网站选项-->
<option value="http://www.sina.com">新浪网站</option>          <!--设置网站选项-->
<option value="http://www.tom.com">TOM 网站</option>          <!--设置网站选项-->
</select>
<a  href="javascript: surfto()" onMouseOver="self.status=";return true"onMouseOut="self.status=";return
true"><img src="ball1.gif"border=0align=absmiddle></a><!--设置图片的链接-->
</form>
……
```

本程序实现了在单击图片链接时，会自动转到指定的网站，难点主要是如何将图片形式转换为按钮。显示的结果如图 12-48 所示。

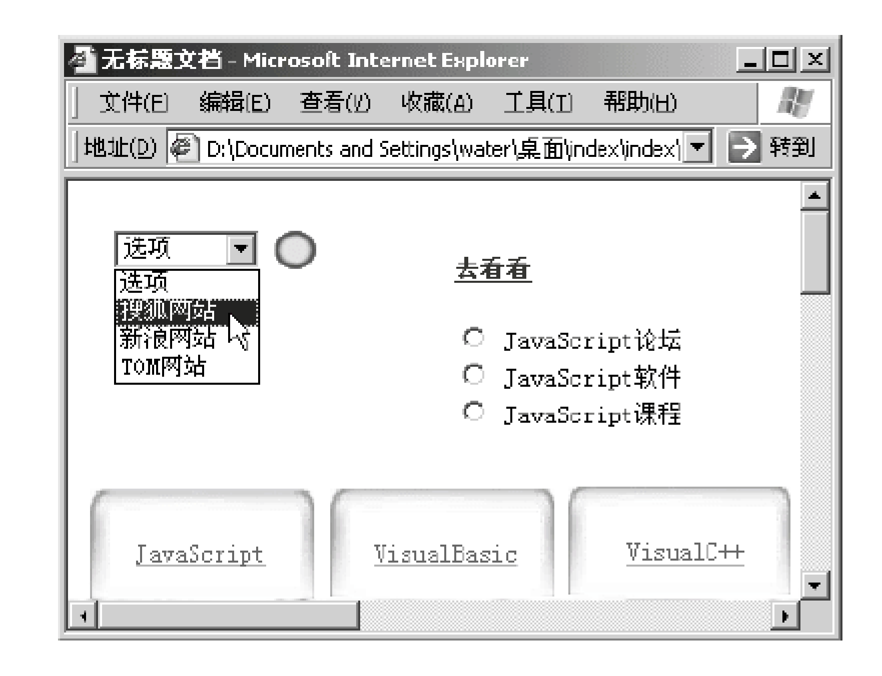


图 12-48　图片代替按钮效果

12.5.15　图片的翻转效果

本实例使用 JavaScript 制作了一个图片的翻转效果，在单击按钮之后，图片就会发生相应的变化。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language=JavaScript">
function setFliph()
{
  <!--设置图片的水平翻转-->
  obj.style.filter=obj.style.filter=="fliph"?" ":"fliph";
}
function setFlipv()
{
  <!--设置图片的垂直翻转-->
  obj.style.filter=obj.style.filter=="flipv"?" ":"flipv";
}
</Script >
<body>
<div ><img src="mouse.jpg"   id="obj"></div>
<form id=form1 name=form1>
```

```
滤镜效果:
<!--设置按钮-->
<input type="button" value="水平翻转" onClick="setFliph()">
<!--设置按钮-->
< input type="button" value="垂直翻转" onClick="setFlipv()">
</form>
……
```

本程序的难点主要是实现在单击按钮之后图片会作出相应的变化，并且当再次单击按钮之后就会还原。显示的结果如图 12-49、图 12-50 和图 12-51 所示。



图 12-49　图片的翻转效果（1）



图 12-50　图片的翻转效果（2）



图 12-51　图片的翻转效果（3）

12.5.16　跟着鼠标的烟花

本实例使用 JavaScript 制作了一个跟随鼠标的烟花效果，当页面被打开后烟花效果就随鼠标不停地旋转。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
var a_Colour='fff000';                          <!--第一个轨迹的颜色-->
var b_Colour='00ff00';                          <!--第二个轨迹的颜色-->
var c_Colour='ff00ff';                          <!--第三个轨迹的颜色-->
var Size=120;
var YDummy=new Array(),XDummy=new
Array(),xpos=0,ypos=0,ThisStep=0;step=0.6;
document.all
{
function ieMouse()
  {
   xpos = document.body.scrollLeft+event.x+6;      <!--获得现在鼠标的横坐标书-->
   ypos = document.body.scrollTop+event.y+16;      <!--获得现在鼠标的纵坐标-->
   }
   document.onmousemove = ieMouse;
}
function swirl()
{
  for (i = 0; i < 3; i++)                          <!--依次处理 3 个轨迹-->
  {
   YDummy[i]=ypos+Size*Math.sin((1*Math.sin((ThisStep)/10))+i*2)*Math.sin((ThisStep)/4);
```

```
                                                      <!--计算得到第 i 个轨迹上每一点的横坐标-->
        XDummy[i]=xpos+Size*Math.cos((1*Math.sin((ThisStep)/10))+i*2)*Math.sin((ThisStep)/4);
                                                      <!--计算得到第 i 个轨迹上每一点的纵坐标-->
    }
    ThisStep+=step;
    setTimeout('swirl()',10);                         <!--周期性调用 swirl ()函数-->
}
var amount=10;
if ( document.all)
{
    document.write('<div id="ODiv"
style="position:absolute;top:0px;left:0px">'+'<div id="IDiv"
style="position:relative">');
    for (i = 0; i < amount; i++)                 <!--依次处理每一个点-->
    {
      document.write('<divid=xstyle="position:absolute;top:0px;left:0px;width:'+i/2+';height:'+i/2+';background:'+
a_Colour+';font-size:'+i/2+'"></div>');          <!--第一个轨迹所在的页面-->
     document.write('<divid=ystyle="position:absolute;top:0px;left:0px;width:'+i/2+';height:'+i/2+';background:'+b
_Colour+';font-size:'+i/2+'"></div>');           <!--第二个轨迹所在的页面-->
     document.write('<divid=zstyle="position:absolute;top:0px;left:0px;width:'+i/2+';height:'+i/2+';background:'+c
_Colour+';font-size:'+i/2+'"></div>');           <!--第三个轨迹所在的页面-->
    }
    document.write('</div></div>');
}
function prepos()
{
    var msie=document.all;                       <!--获得 IE 浏览器的当前页面-->
    if(document.all)
    {
      for (i = 0; i < amount; i++)               <!--依次处理每一个点-->
      {
          if (i < amount-1)                      <!--对于前 amount-1 个点-->
          {
                msie.x[i].style.top=msie.x[i+1].style.top;msie.x[i].style.left=msie.x[i+1].style.left;
                <!--更新第一个轨迹上各个点上的上边界和左边界-->
                msie.y[i].style.top=msie.y[i+1].style.top;msie.y[i].style.left=msie.y[i+1].style.left;
                <!--更新第二个轨迹上各个点上的上边界和左边界-->
                msie.z[i].style.top=msie.z[i+1].style.top;msie.z[i].style.left=msie.z[i+1].style.left;
                <!--更新第三个轨迹上各个点上的上边界和左边界-->
     }
     else
     {
                <!--更新第一个轨迹上最后一个点上的上边界和左边界-->
                msie.x[i].style.top=YDummy[0];msie.x[i].style.left=XDummy[0];
                <!--更新第二个轨迹上最后一个点上的上边界和左边界-->
                msie.y[i].style.top=YDummy[1];msie.y[i].style.left=XDummy[1];
                <!--更新第三个轨迹上最后一个点上的上边界和左边界-->
                msie.z[i].style.top=YDummy[2];msie.z[i].style.left=XDummy[2];
       }
```

```
        }
    }
    setTimeout("prepos()",10);                    <!--周期性调用 prepos(函数)-->
}
function Start()                                  <!--开始(函数)-->
{
    swirl(),prepos()                              <!--依次调用 swirl 和 prepos(函数)-->
}
window.onload=Start;                              <!--调用 start(函数)-->
</script>
……
```

本程序的难点主要是实现跟着鼠标的烟花的效果，以及确定的动画轨迹的实现方法。显示的结果如图 12-52 所示。



图 12-52　跟着鼠标的烟花

12.5.17　跟随鼠标的时钟

本实例使用 JavaScript 制作了一个跟随鼠标的时钟，当打开页面之后，在鼠标的右面会有一个时钟来显示时间。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
colors = new Array('330099','00ff00','ff00ff');
sCol = 'ff00ff';
mCol = '00ff00';
hCol = 'red';
H = '***';
H = H.split(' ');
H = H.reverse();
M = '*****';
M = M.split(' ');
M = M.reverse();
S = '*****';
S = S.split(' ');
S = S.reverse();
```

```
    dots = 12;
    var Ypos = 0,Xpos = 0,Ybase = 0,Xbase = 0;
    var ay = 0, ax = 0, Ay = 0, Ax = 0, by = 0, bx = 0, By = 0, Bx = 0, cy= 0,
cx = 0, Cy = 0, Cx = 0, dy = 0, dx = 0, Dy = 0, Dx = 0;
    count = 0;
    count_a = 0;
    move = 1;
    viz ='hidden';
    if (document.all)
    {
      document.write('<div id="W" style="position:absolute;top:0px;left:0px"><div
    style="position:relative">');
      for (i = 0; i < dots; i++)
      {
       document.write('<div id="face"
    style="position:absolute;top:0px;left:0px;width:3px;height:3px;font-size:3px
    ;background:#000099"></div>');
    }
    document.write('</div></div>');
    document.write('<div id="X"
    style="position:absolute;top:0px;left:0px"><div
    style="position:relative">');
      for (i = 0; i < S.length; i++)
      {
       document.write('<divid="x"style="position:absolute;width:36px;height:36px;font-family:Verdana;font-size:12p
    x;color:'+sCol+';text-align:center;padding-top:10px">'+S[i]+'</div>');
    }
       document.write('</div></div>')
       document.write('<div id="Y" style="position:absolute;top:0px;left:0px"><div
    style="position:relative">');
    for (i = 0; i < M.length; i++)
    {
      document.write('<divid="y"
    style="position:absolute;width:36px;height:36px;font-family:Verdana;font-size:12px;color:'+mCol+';text-align:
    center;padding-top:10px">'+M[i]+'</div>');
    }
      document.write('</div></div>')
      document.write('<div id="Z"
    style="position:absolute;top:0px;left:0px"><div
    style="position:relative">');
    for (i = 0; i < H.length; i++)
    {
      document.write('<divid="z"
    style="position:absolute;width:36px;height:36px;font-family:Verdana;font-size:12px;color:'+hCol+';text-align:
    center;padding-top:10px">'+H[i]+'</div>');
    }
      document.write('</div></div>');
    }
    if (document.all)
```

```
{
  function ieMouse()
  {
     Ypos = event.y + 100;
     Xpos = event.x + 100;
 }
 document.onmousemove = ieMouse;
}
function clock()
{
   time = new Date ();
   secs = time.getSeconds();
   sec = -1.57 + Math.PI * secs / 30;
   mins = time.getMinutes();
   min = -1.57 + Math.PI * mins / 30;
   hr = time.getHours();
   hrs = -1.575 + Math.PI * hr / 6 + Math.PI * parseInt(time.getMinutes())/ 360;
   Ybase = 15;
   Xbase = 15;
   if (document.all)
   {
     var scrll = document.body.scrollTop;
     W.style.pixelTop = scrll;
     X.style.pixelTop = scrll;
     Y.style.pixelTop = scrll;
     Z.style.pixelTop = scrll;
     x[0].style.visibility=viz;
     y[0].style.visibility = viz;
     z[0].style.visibility = viz;
     for (i = 0; i < S.length; i++)
     {
       x[i].style.pixelTop = ay - 12 + (i * Ybase) * Math.sin(sec);
       x[i].style.pixelLeft = ax - 12 + (i * Xbase) * Math.cos(sec);
}
for (i = 0; i < M.length; i++)
{
       y[i].style.pixelTop = by - 12 + (i * Ybase) * Math.sin(min);
       y[i].style.pixelLeft = bx - 12 + (i * Xbase) * Math.cos(min);
}
for (i = 0; i < H.length; i++)
{
       z[i].style.pixelTop = cy - 12 + (i * Ybase) * Math.sin(hrs);
       z[i].style.pixelLeft = cx - 12 + (i * Xbase) * Math.cos(hrs);
}
for (i = 0; i < dots; ++i)
{
  face[i].style.pixelTop = dy + 6 + (70 * Math.sin(-0.49 + dots + i /1.9));
  face[i].style.pixelLeft = dx + 4 + (70 * Math.cos(-0.49 + dots + i /1.9));
    }
```

```
    }
  }
  function MouseFollow()
  {
    ay = Math.round(Ay += ((Ypos) - Ay) * 4 / 15);
    ax = Math.round(Ax += ((Xpos) - Ax) * 4 / 15);
    by = Math.round(By += (ay - By) * 4 / 15);
    bx = Math.round(Bx += (ax - Bx) * 4 / 15);
    cy = Math.round(Cy += (by - Cy) * 4 / 15);
    cx = Math.round(Cx += (bx - Cx) * 4 / 15);
    dy = Math.round(Dy += (cy - Dy) * 4 / 15);
    dx = Math.round(Dx += (cx - Dx) * 4 / 15);
    clock();
    setTimeout('MouseFollow()',10);
  }
  function StartAll()
  {
    MouseFollow();
  }
  if (document.layers || document.all) window.onload = StartAll;
  </Script>
  ……
```

本程序的难点主要是跟随鼠标时钟的实现，并且时钟的时间要和本地时间保持一致。显示的结果如图 12-53 所示。



图 12-53　跟随鼠标的时钟

12.5.18　跟随鼠标的滚动字幕

本实例使用 JavaScript 制作了一个跟随鼠标的滚动字幕，在滚动框内可以设置一些问候语或提示文字。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
<!--设置显示的内容-->
var scroller_msg='<font color=#FF00FF>欢迎您光临本站</font>'
<!--设置多长时间后才显示滚动的内容-->
var dismissafter=0
var initialvisible=0
if (document.all)
document.write('<marquee id="curscroll"
style="position:absolute;width:100px;border:1px solid
black;font-size:14px;background-color:white;visibility:hidden">'+scroller_msg+'</marquee>')
                                                            <!--设置滚动框样式-->
function followcursor()
{
  if (initialvisible==0)
  {
```

```
            curscroll.style.visibility="visible"
            initialvisible=1
        }
        <!--设置 X 轴位置-->
        curscroll.style.left=document.body.scrollLeft+event.clientX+10
        <!--设置 Y 轴位置-->
        curscroll.style.top=document.body.scrollTop+event.clientY+10
    }
    function dismissmessage()
    {
      curscroll.style.visibility="hidden"
    }
    if (document.all)
    {
      document.onmousemove=followcursor
      document.ondblclick=dismissmessage
      if (dismissafter!=0)
      setTimeout("dismissmessage()",dismissafter*1000)
    }
    </Script>
    ……
```

本程序的难点主要是实现跟随鼠标的提示框，并且设置提示框内的文字可以以滚动的方式出现。显示的结果如图 12-54 所示。



图 12-54　跟随鼠标的滚动字幕

12.5.19　不断闪动的链接

本实例使用 JavaScript 制作了一个文字闪动的链接，当单击链接时就链接到设置好的 URL。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
function initArray()                                   <!--定义闪烁的颜色变化顺序-->
{
  <!--循环变量初始化和依次递加-->
```

```
    for (var i=0; i < initArray.arguments.length;i++)
     {
      this[i]=initArray.arguments[i];                            <!--变量赋值-->
     }
     this.length=initArray.arguments.length;                     <!--记录颜色数组的长度-->
  }
  varcolors=new initArray("#000000","#0000FF","#80FFFF","#80FF80","#FFFF00","#FF8000","#
  FF00FF","#FF0000");                                            <!--变幻时的颜色-->
    delay=100                                                    <!--定义每种颜色闪烁的时间-->
    link=0;                                                      <!--初始化循环变量-->
    vlink=0;
    function linkDance()
  {
    link=(link+1)%colors.length;
    vlink=(vlink+1)%colors.length;                               <!--通过取整运算实现循环-->
    document.linkColor=colors[link];
    <!--将颜色取值分别赋给链接颜色和下划线颜色数组-->
    document.vlinkColor=colors[vlink];
    setTimeout("linkDance()",delay);                             <!--延迟 delay 的时间长度-->
    }
    linkDance();                                                 <!--调用 linkDance()函数实现闪烁功能-->
  </Script>
  <p align="center"><font size="10" face="黑体"><a href="#" target="_blank">学生成绩查询</a></font></p>
  ……
```

本程序的难点主要是实现超链接的闪烁功能，并在程序中设定了闪烁时间的长短，色彩顺序等。在程序中主要是通过调用 linkDance()函数来实现闪烁功能的。显示的结果如图 12-55 和图 12-56 所示。

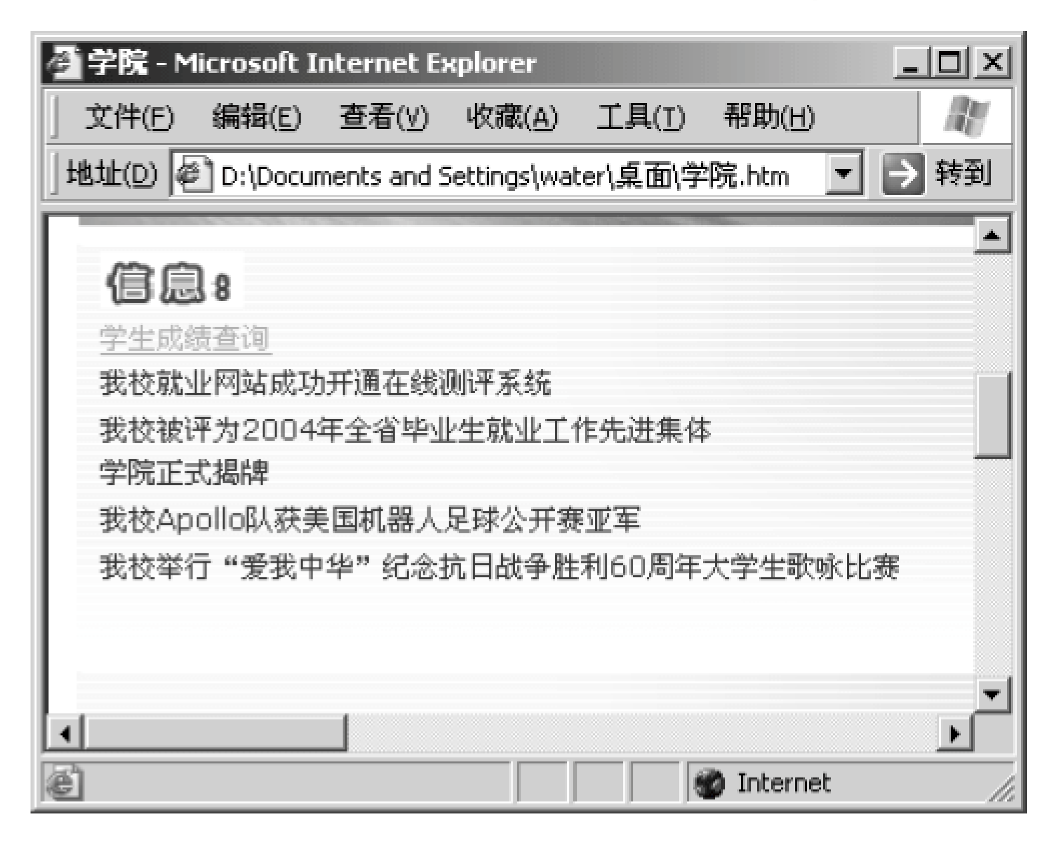


图 12-55　链接闪动效果（1）



图 12-56　链接闪动效果（2）

12.5.20　在按钮上定时显示不同的链接

本实例使用 JavaScript 制作了一个按钮的链接，按钮会随时间的变化而变化，并且每个不同名称的按钮会有不同的链接。下面就来看一下这个实例的具体实现步骤。

```
……
<Script Language="JavaScript">
var startTime=null;
var timerID=null;
var initial=new Date();
var pos=0;
var menuItem=null;
function initArray()                                   <!--矩阵生成函数-->
{
  this.length=initArray.arguments.length
   for (var i=0; i < this.length;i++)
{
  <!--根据输入的字符串，构造一个矩阵-->
  this[i+1] = initArray.arguments[i]
 }
}
function parsemenuItem(data,num)            <!--将数组中每一项中的内容和 url 分开-->
{
  for(var i=0;i<data.length;i++)
  {
    if(data.substring(i,i+1)=="|")   break;                <!--找到分隔符-->
  }
    if (num==0) return(data.substring(0,i));
    else   return(data.substring(i+1,data.length));
}
function startTimer()
{
  initial = new Date();                               <!--获得当前日期-->
  startTime=initial.getTime();
  stopTimer();                                        <!--清除目前的定时器-->
   menuItem = new initArray(
     "南京大学|#",
     "北京大学|#",
     "清华大学|#",
     "复旦大学|#",
     "浙江大学|#");
    showTimer();                                        <!--调用 shouTimer()函数-->
}
function stopTimer()                                    <!--定时停止函数-->
 {
   timerID=null;                                        <!--清空定时器-->
   menuItem=null;                                       <!--清空定时器-->
 }
function showTimer()                                    <!--循环显示按钮内容的函数-->
{
  pos= (pos == menuItem.length) ? 1 : pos + 1;          <!--更新 pos 的值-->
  <!--更新显示的链接-->
 document.forms[0].elements[0].value=parsemenuItem(menuItem[pos], 0);
 <!--每 1 秒调用一次 showTimer()函数-->
```

```
    timerID=window.setTimeout('showTimer()',1000);
  }
  function goToUrl()
  {
    <!--获取当前显示对应的链接，并返回-->
    this.location=parsemenuItem(menuItem[pos],1);
    return (false);
  }
  </Script>
  <body background="06.jpg" onLoad="window.startTimer()">
  <form>
  <p align="center">
  <input type="button" value="WHERE TO?"
name="goTo"onClick="window.goToUrl()">
  </form>
  ……
```

本程序的难点主要是实现在按钮上定时显示不同的链接功能，并在程序中使用了按钮控件、数组、字符串的相关方法，以及定时器的使用。显示的结果如图 12-57 和图 12-58 所示。

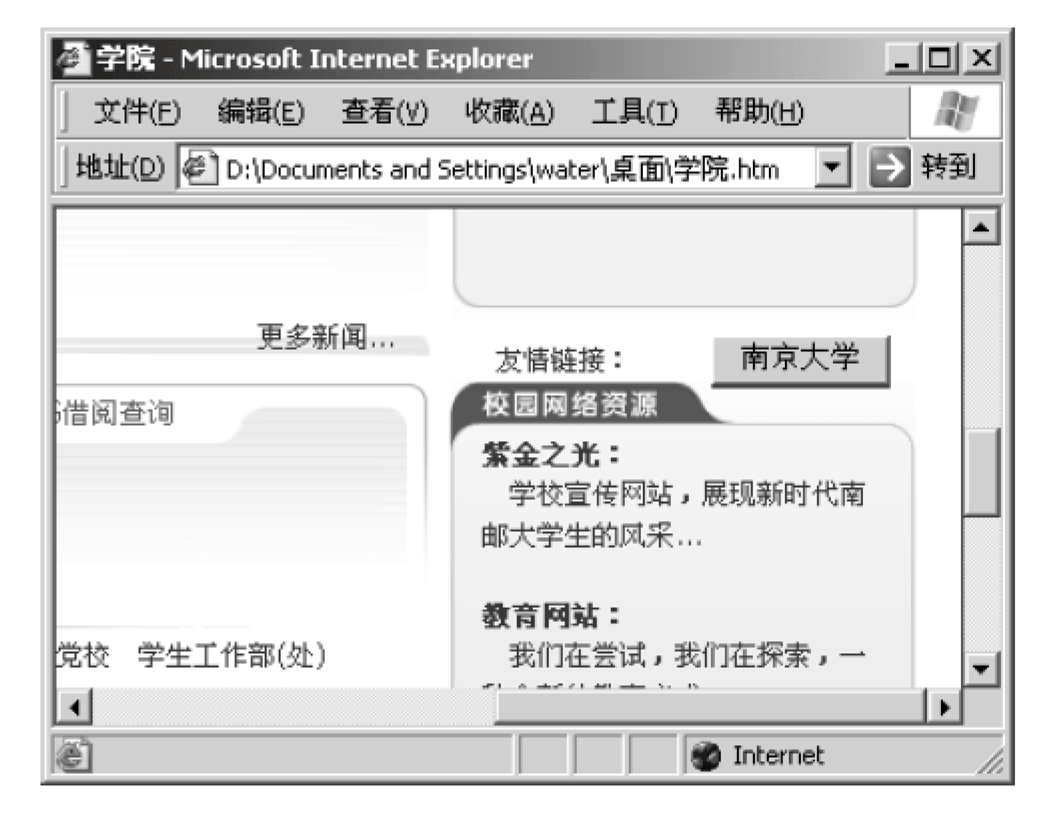


图 12-57　按钮的链接效果（1）

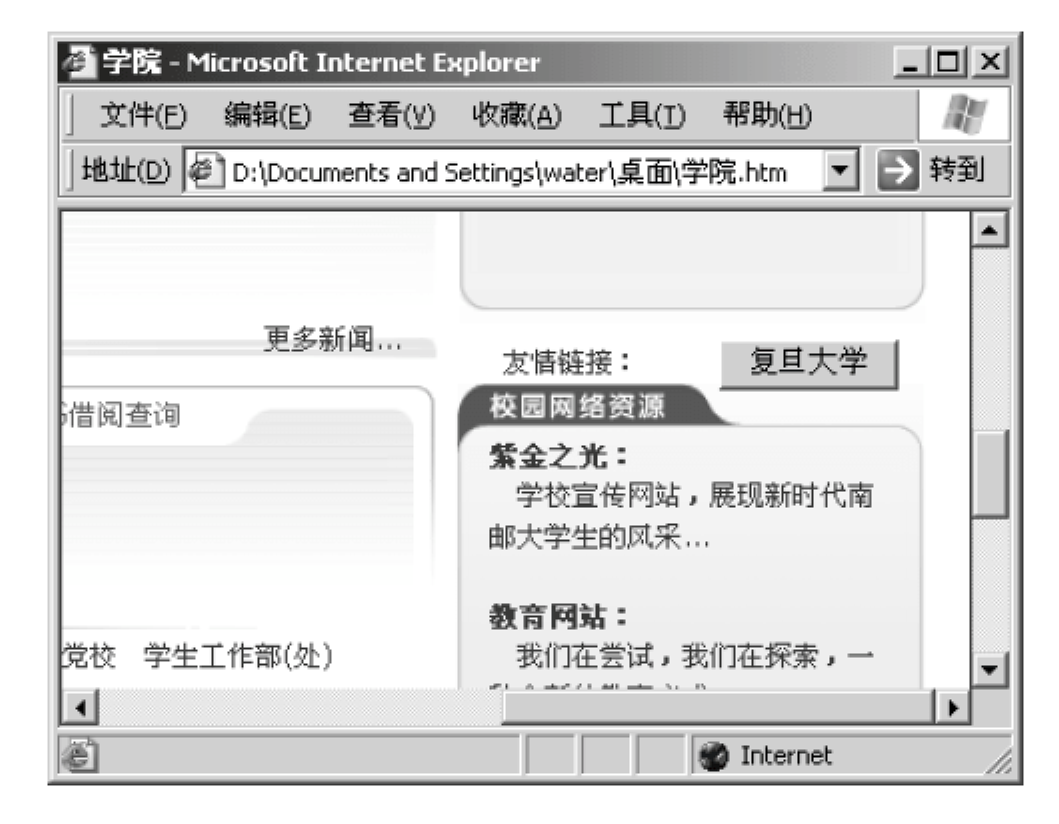


图 12-58　按钮的链接效果（2）

12.5.21　带链接的滚动字幕

本实例使用 JavaScript 制作了一个滚动字幕的链接，字幕会从屏幕的右边向左滚动，并在指定的字体上添加链接。下面就来看一下这个实例的具体实现步骤。

```
  ……
  <Script Language="JavaScript">
  var marqueewidth=400                    <!--设置 marquee 的宽度-->
  var marqueeheight=20                    <!--设置 marquee 的高度-->
  var speed=5                             <!--设置 marquee 滚动的速度-->
  <!--设置 marquee 显示内容，使用标准的 HTML 语法。-->
  var marqueecontents='<strong><big> <a href="http://www.xxx.com">我校 Apollo
队获美国机器人足球公开赛亚军</a></big></strong></font>'
  document.all
```

```
    document.write('<marquee scrollAmount='+speed+'
style="width:'+marqueewidth+'">'+marqueecontents+'</marquee>')
    </Script>
    ……
```

本程序的难点主要是实现字幕滚动的链接功能，并在程序中设置滚动的宽度、高度和速度等参数，并为指定的字体添加链接。显示的结果如图 12-59 和图 12-60 所示。



图 12-59　滚动的链接字幕（1）

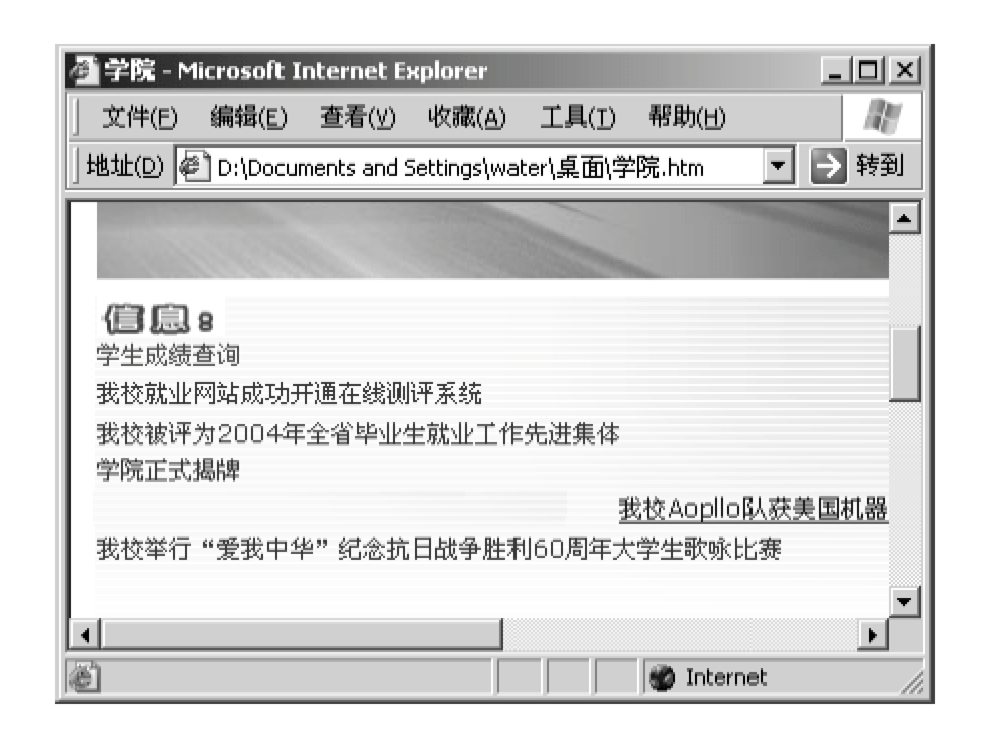


图 12-60　滚动的链接字幕（2）

第13章 在 Dreamweaver 中创建网页

- 常用页面元素
- 网页的布局
- 表单的使用
- 添加各种类型的文本
- 添加 HTML 网页的辅助内容

Dreamweaver 是一个可视化的网页设计和网站管理工具。它提供了强大的设计工具，在不用书写一行代码的情况下，就能够快速创建各种极具动态 HTML 特性的网页，例如动画和分层等。在可视编辑器中进行编辑时，可以同步看到 Dreamweaver 生成的源代码。

Dreamweaver 是完全可定制的，用户可以创建自己的对象和命令，修改菜单和快捷键，还可以通过书写 JavaScript 代码为 Dreamweaver 创建新的行为和属性面板，以增强 Dreamweaver 本身的能力。

由于 Dreamweaver 的功能强大，本章以 Dreamweaver MX 2004 为例，只介绍具有代表性的 HTML 元素在 Dreamweaver 中的添加方式。

13.1　常用页面元素

常用页面元素主要包括在页面中导入图像、多媒体元素、表格等。

13.1.1　插入图像

在 Dreamweaver 中插入图像不需要手工编写代码，具体实现步骤如下：

（1）启动 Dreamweaver MX 2004。

（2）选择“文件”|“新建”命令打开“创建文档”对话框。激活“常规”选项卡，选中“基本页”列表框中的 HTML，如图 13-1 所示。单击“创建”按钮创建一个新的 HTML 页面。

（3）在文档窗口中选择“拆分”视图的显示模式，此时可以同时显示 HTML 文件的代码和页面设计效果，如图 13-2 所示。



图 13-1　新建一个 HTML 文档

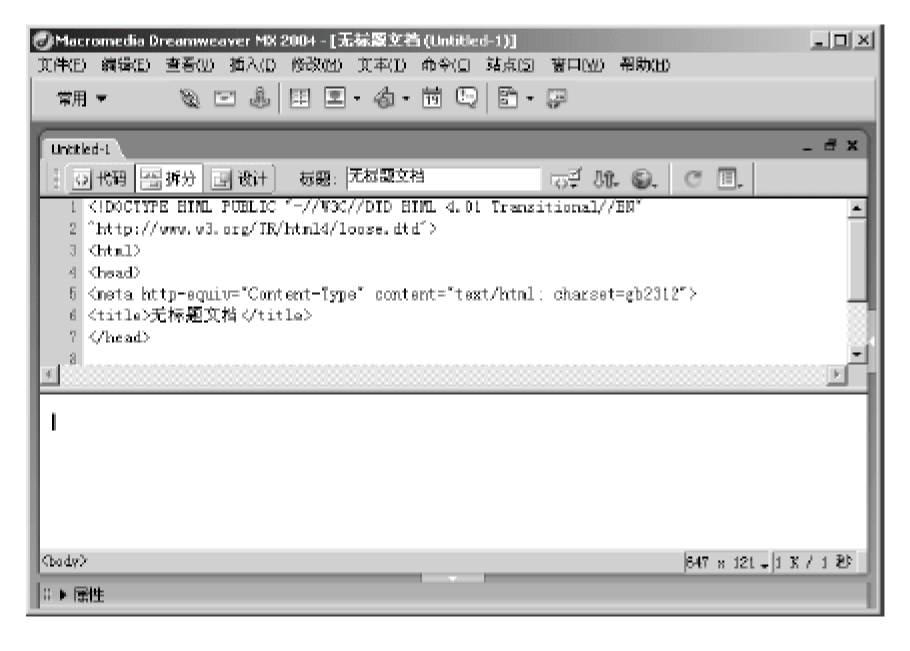


图 13-2　“拆分”视图显示页面

（4）选择“文件”|“保存”命令，将 HTML 文件保存。

（5）为了便于操作和演示，此处将文档的视图设置为“设计”模式。选择“插入”|“图像”命令或者直接单击“常用”工具栏中的“插入图像”按钮，打开“选择图像源文件”对话框，如图 13-3 所示。



图 13-3　“选择图像源文件”对话框

（6）在对话框中选择要插入的图像文件，单击“确定”按钮将图像插入到 HTML 页面中，效果如图 13-4 所示。

（7）在页面中的“属性”面板中可以调整图像的各种属性。在“宽”和“高”文本框中可以设置图像的大小，如果只设置其中的一个值，图像会等比例缩放，效果如图 13-5 所示。

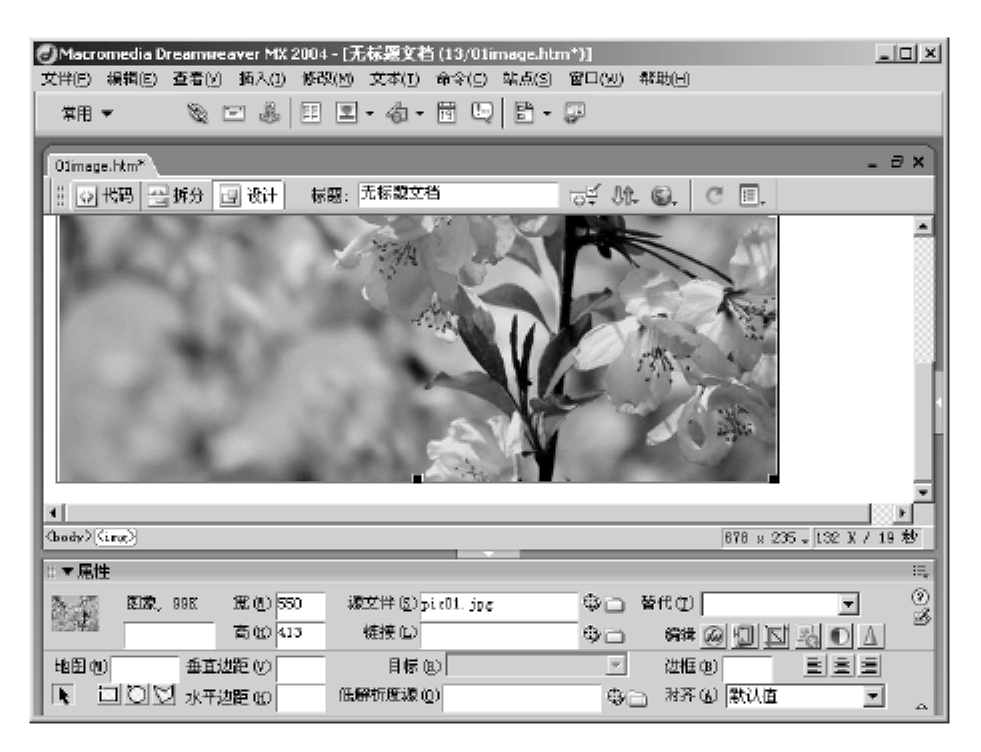


图 13-4　插入图像文件

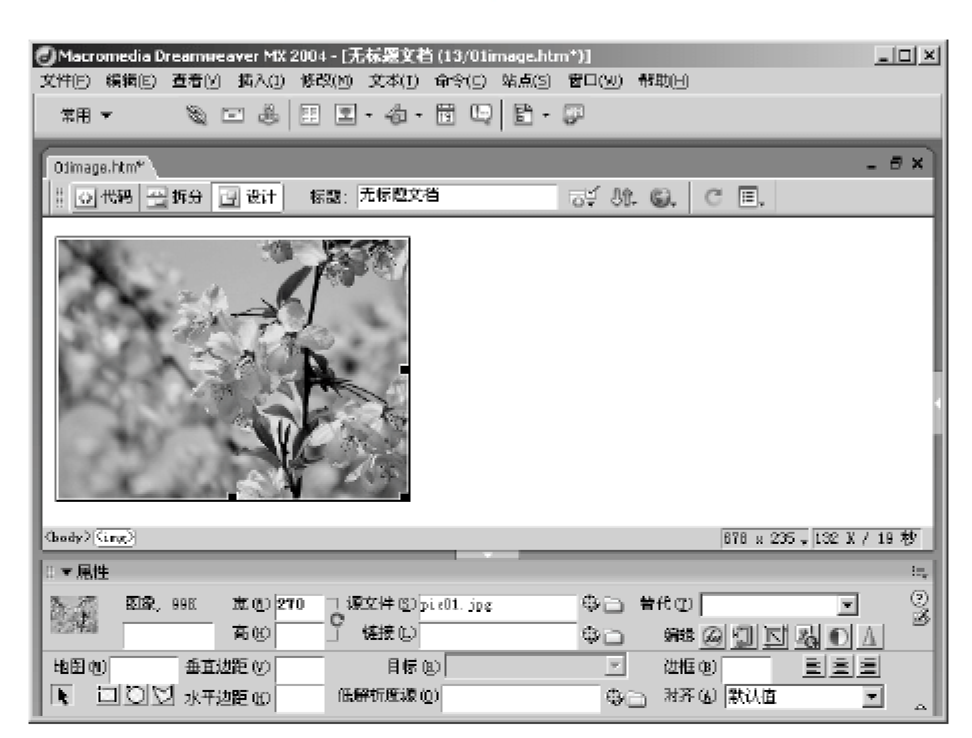


图 13-5　调整图像大小

（8）此外，在最左侧的“图像”文本框中可以为图像命名；在“源文件”文本框中可以直接调整图像文件的地址；在“替代”文本框中可以设置图像的提示文字；在“链接”文本框中可以设置图像的链接地址。本实例中设置图像的各种参数如图 13-6 所示。

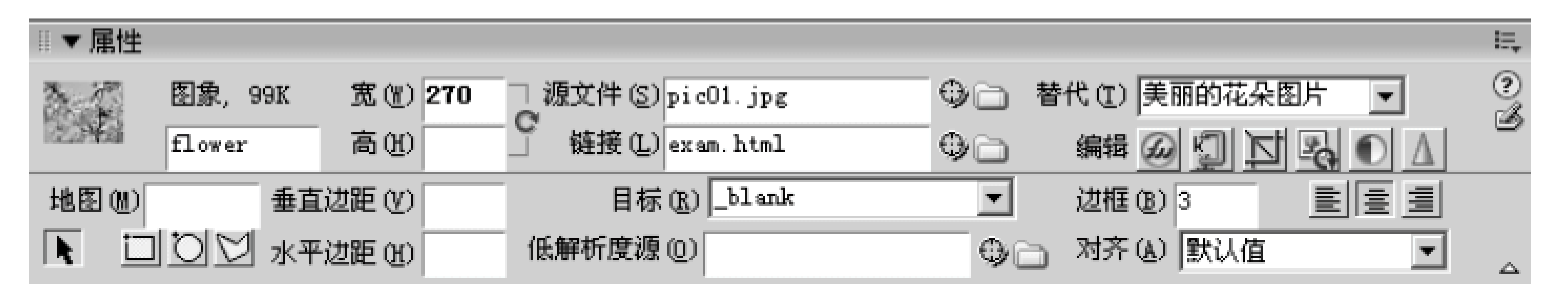


图 13-6　调整图像的其他属性

（9）选择“文件”|“保存”命令，保存 HTML 文档的内容，在 Internet Explorer 中打开文件，效果如图 13-7 所示。单击页面中的图像，会打开一个新的链接页面，效果如图 13-8 所示。

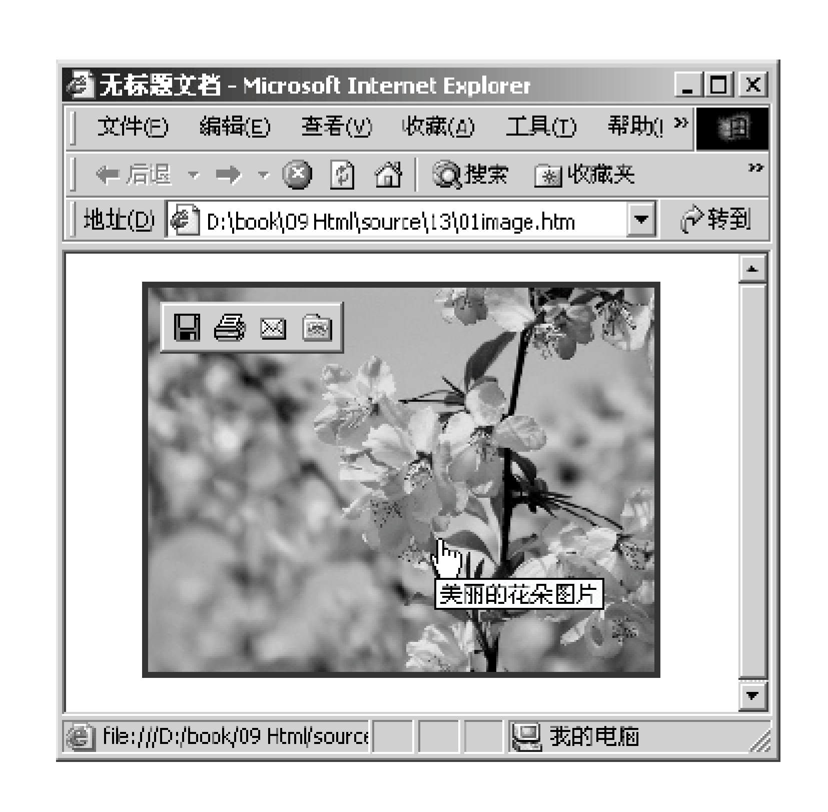


图 13-7　使用 Dreamweaver 插入图像文件的效果



图 13-8　打开链接页面

13.1.2　导入媒体

媒体导入的方法与图像的导入方法类似，具体操作步骤如下：

（1）新建一个 HTML 文件并保存。

（2）选择“插入”|“媒体”命令或单击“常用”工具栏中的“媒体”按钮，在其下拉列表框中选择要插入的媒体类型，此处选择“Flash 按钮”，此时会弹出一个“插入 Flash 按钮”对话框。

（3）在对话框中可以设置按钮的样式、按钮上的文字内容、文字字体、按钮的链接地址等，此处设置 Flash 按钮的各种属性如图 13-9 所示。

（4）保存并运行页面文档，效果如图 13-10 所示。

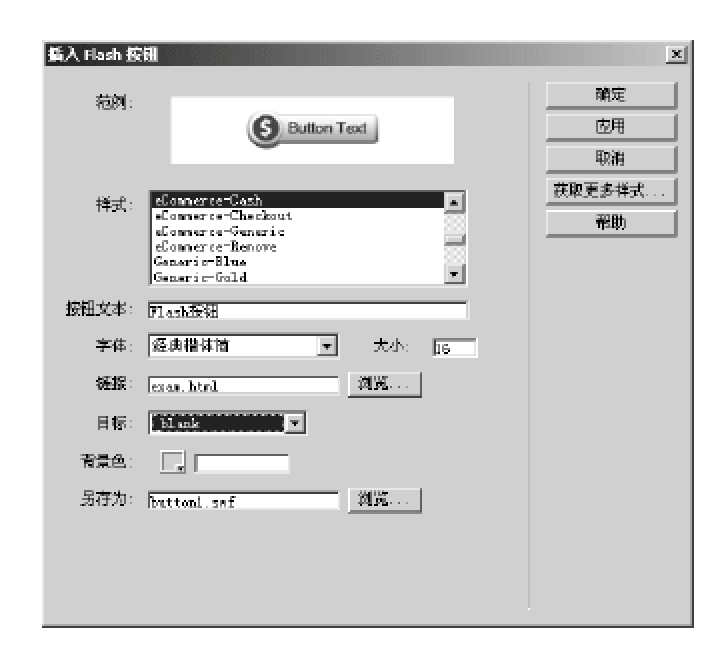


图 13-9　“插入 Flash 按钮”对话框



图 13-10　添加 Flash 按钮的效果

（5）添加其他媒体文件的方法与此类似，此处不再重复讲解。

13.1.3　添加表格

在 Dreamweaver 中为 HTML 页面添加表格的操作步骤如下：

（1）新建一个 HTML 文件并保存。

（2）选择“插入”|“表格”命令或直接单击“常用”工具栏中的“插入表格”按钮，打开“表格”对话框。

（3）在该对话框中可以设置表格的大小、宽度、边框属性、边距属性等各种表格参数。此处设置表格的各种属性如图 13-11 所示。

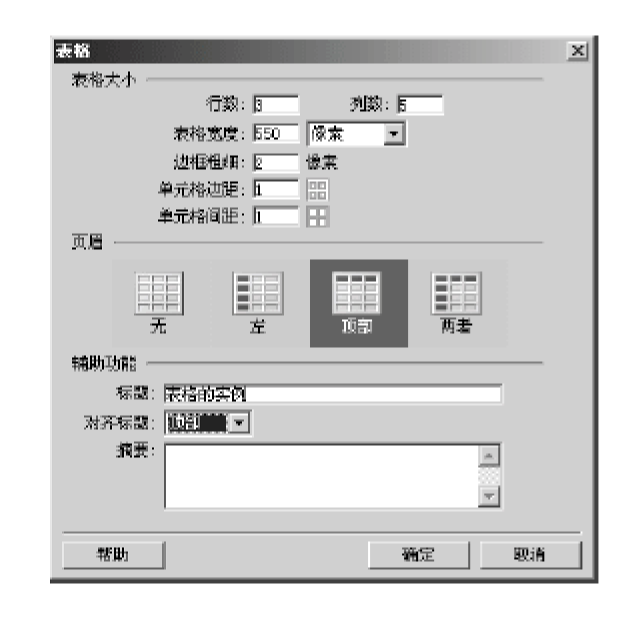


图 13-11　设置表格的各种属性

（4）在“属性”面板中可以对表格的背景、边框颜色等进行调整，如图 13-12 所示。

（5）选中表格中的某一行还可以直接调整这一行的属性，如图 13-13 所示。

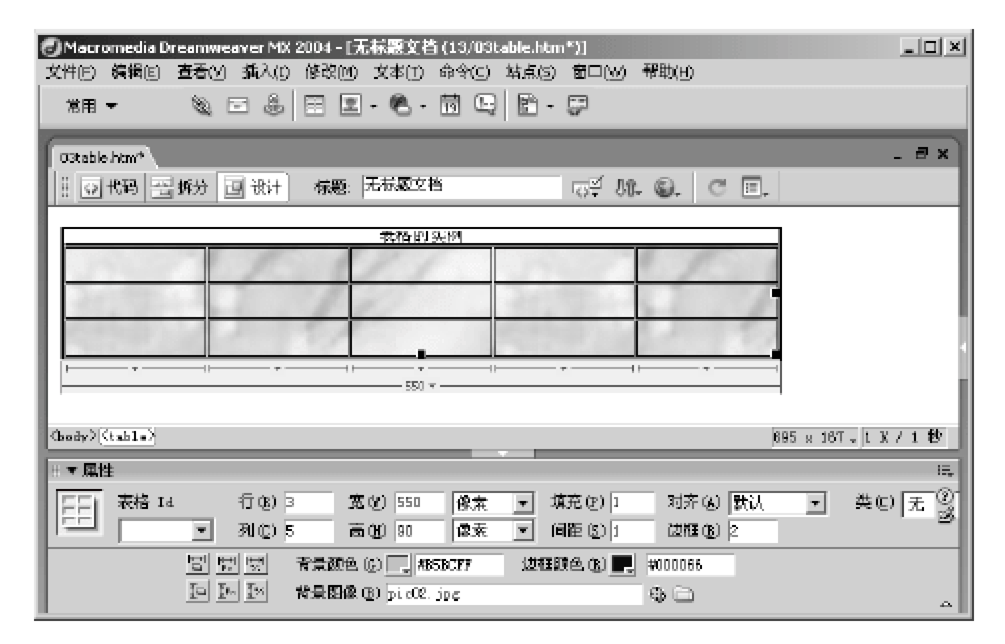


图 13-12　调整表格的属性



图 13-13　调整一行表格的属性

（6）同样的方法，如果选中其中某一个单元格，可以对该单元格进行单独设置，如图 13-14 所示。

（7）在表格中可以直接添加文本内容，并对文本字体进行调整，如图 13-15 所示。

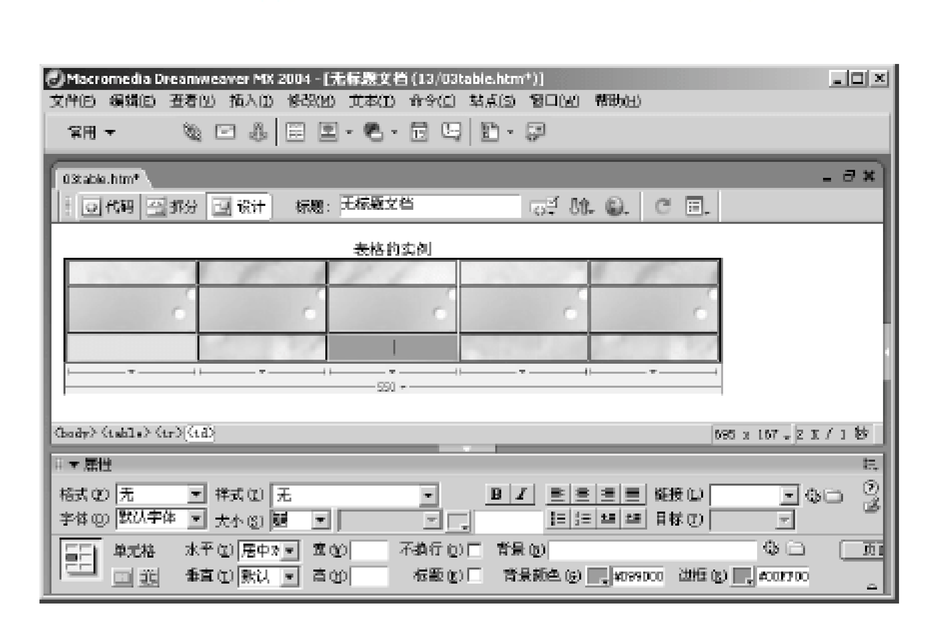


图 13-14　设置单元格属性



图 13-15　在表格中添加文字

（8）完成文字的添加后即可保存文件。

13.1.4　添加时间元素

Dreamweaver 还提供了为页面添加时间元素的功能，添加方法如下：

（1）新建一个 HTML 文件，为该文件命名并保存。

（2）选择“插入”|“日期”命令或直接单击“日期”按钮，打开“插入日期”对话框。

（3）在对话框中可以设置日期和时间的格式，如图 13-16 所示。

（4）保存文件修改后运行程序，可以看到页面上显示了时间，效果如图 13-17 所示。



图 13-16　设置日期和时间的格式



图 13-17　运行程序的效果

13.1.5　设置超级链接

在 Dreamweaver 中添加超级链接很方便，具体操作步骤如下：

（1）新建一个文件，为该文件命名并保存。

（2）单击“常用”工具栏中的 “超级链接”按钮，打开 “超级链接”对话框，如图 13-18 所示。

（3）在对话框中可以设置要进行链接的文本、链接地址、目标窗口的打开方式等，此处设置效果如图 13-19 所示。

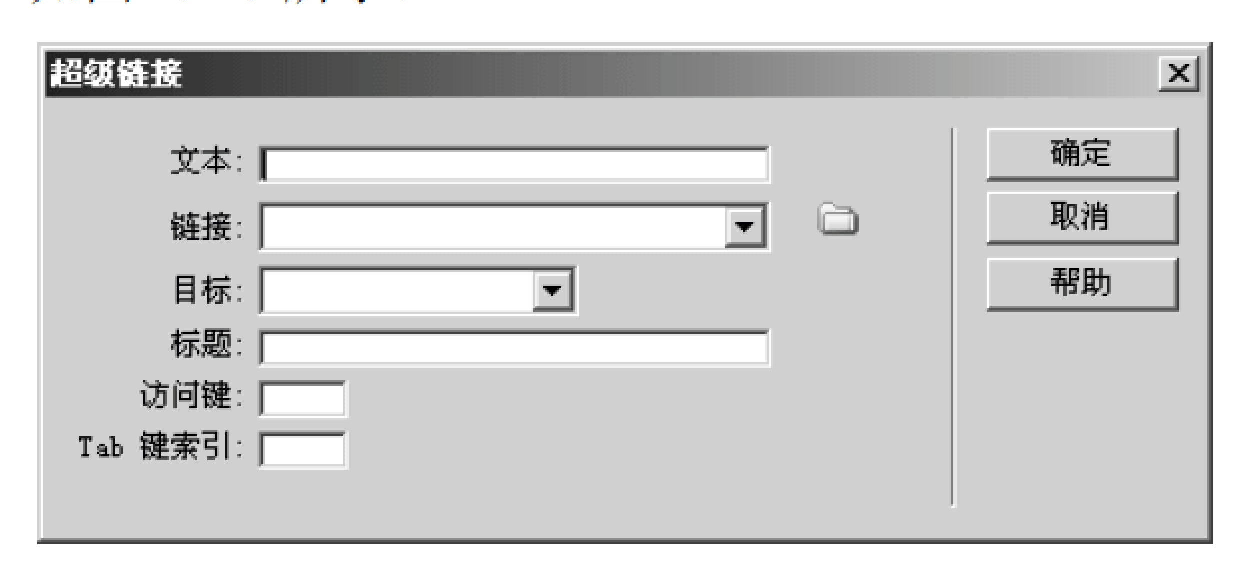


图 13-18　“超级链接”对话框

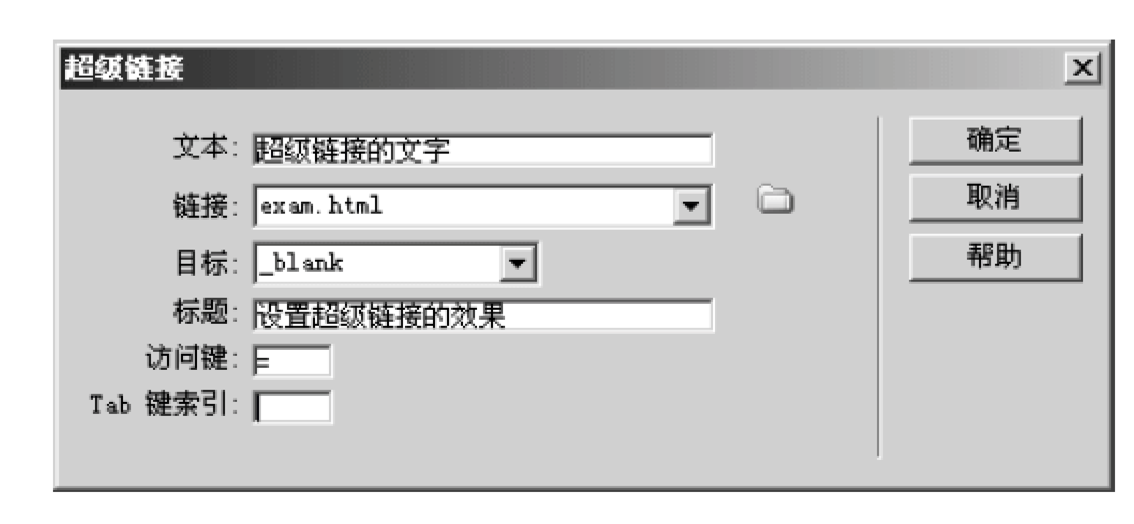


图 13-19　设置超级链接

（4）保存文件后运行程序，将鼠标移动到链接文字上方会出现标题文字“设置超级链接的效果”，单击链接文字会打开链接窗口页面，效果如图 13-20 所示。



图 13-20　添加超级链接的效果

13.2　网页的布局

页面布局主要是通过表格、框架和层来实现，在 Dreamweaver 中可以轻松实现页面的布局。

13.2.1　使用布局表格

使用布局表格的方法如下：

（1）新建一个 HTML 文件，为该文件命名并保存。

（2）单击工具栏中的常用▼，在下拉列表中选择“布局”，在工具栏中显示布局类的按钮，如图 13-21 所示。



图 13-21　“布局”工具栏

（3）单击工具栏中的“布局”按钮布局，弹出如图 13-22 所示的提示窗口。单击“确定”按钮进入布局模式。

（4）单击“布局表格”按钮，可以在页面中绘制布局表格，默认情况下，布局表格是从页面的左上角开始的，如图 13-23 所示。

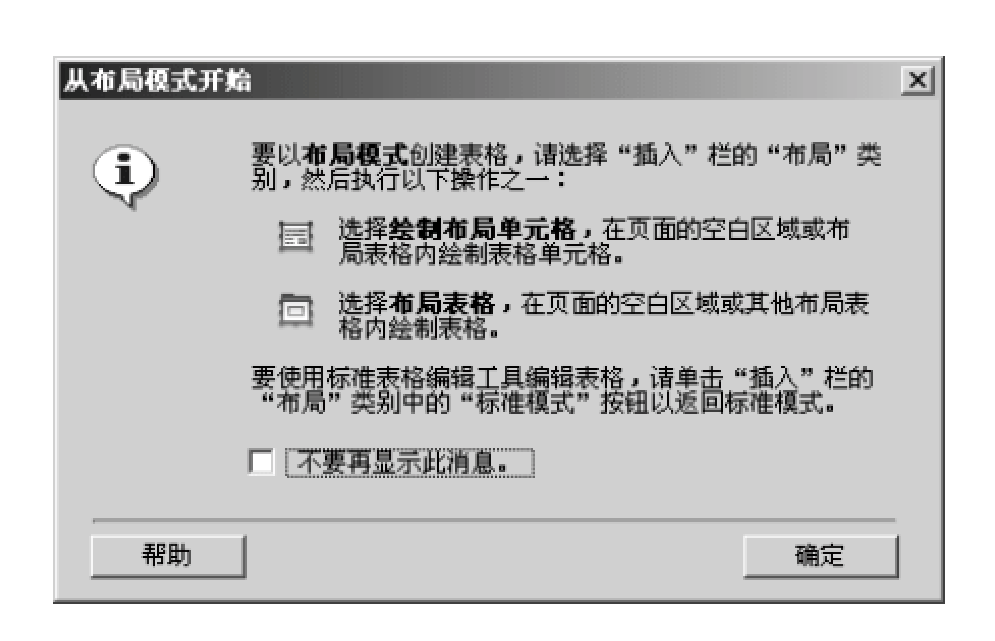


图 13-22　转换为布局模式的提示窗口

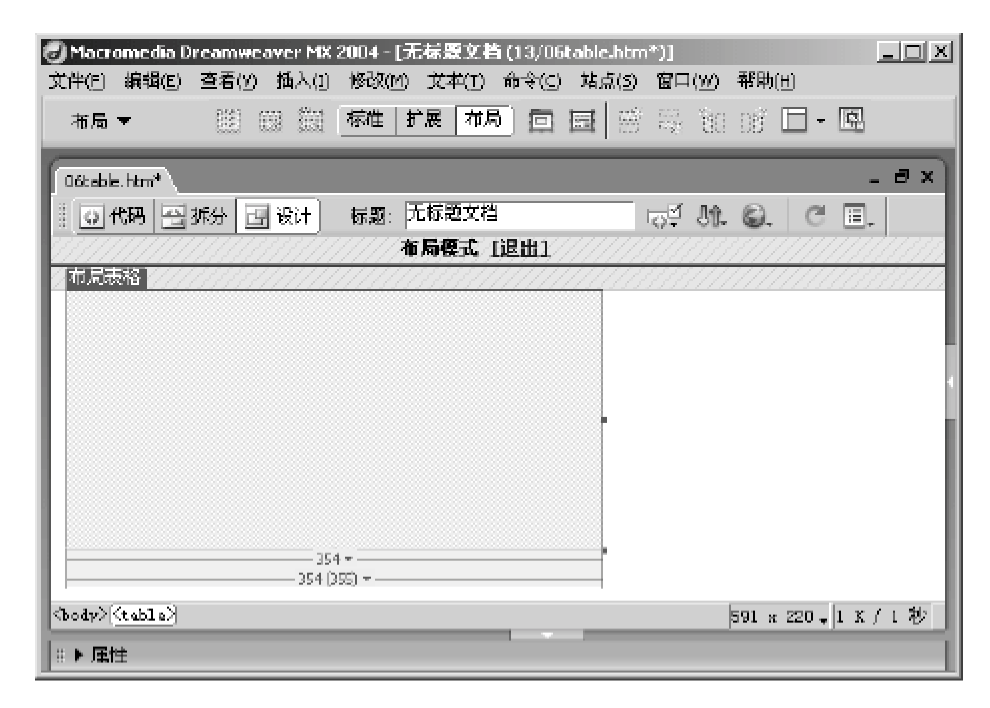


图 13-23　绘制布局表格

（5）单击工具栏中的“绘制布局单元格”按钮，在布局表格中可以绘制单元格，如图 13-24 所示。

（6）单击文档窗口中的“退出”按钮，可以退出布局模式，此时页面会自动将刚才创建的页面布局保存为表格的形式，效果如图 13-25 所示。

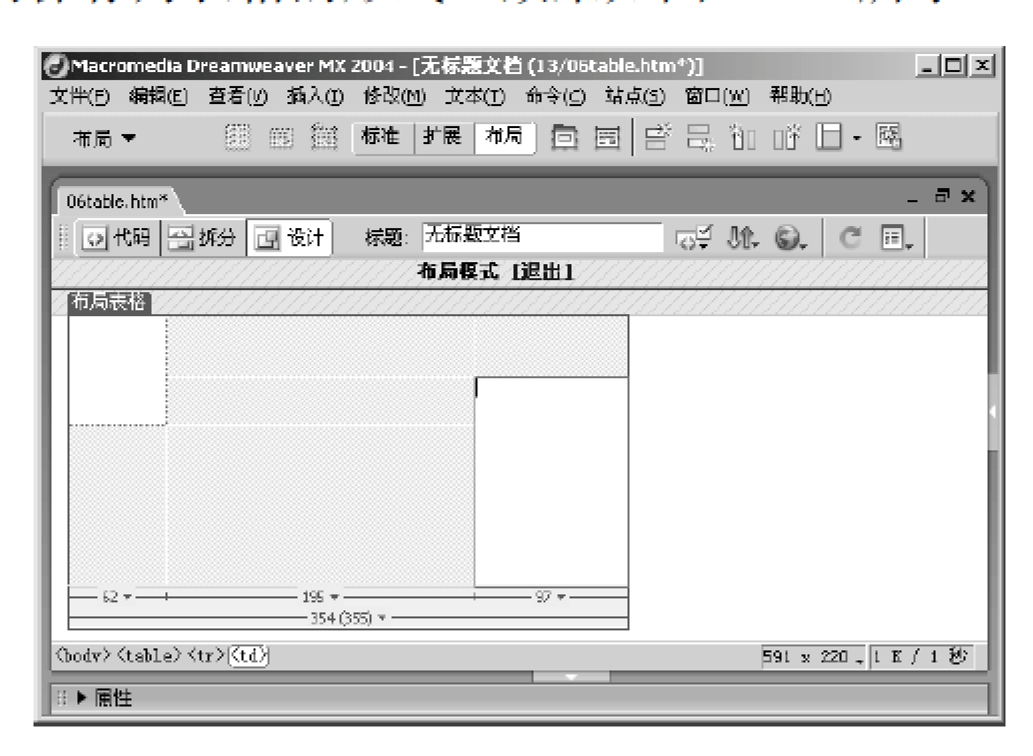


图 13-24　创建布局单元格

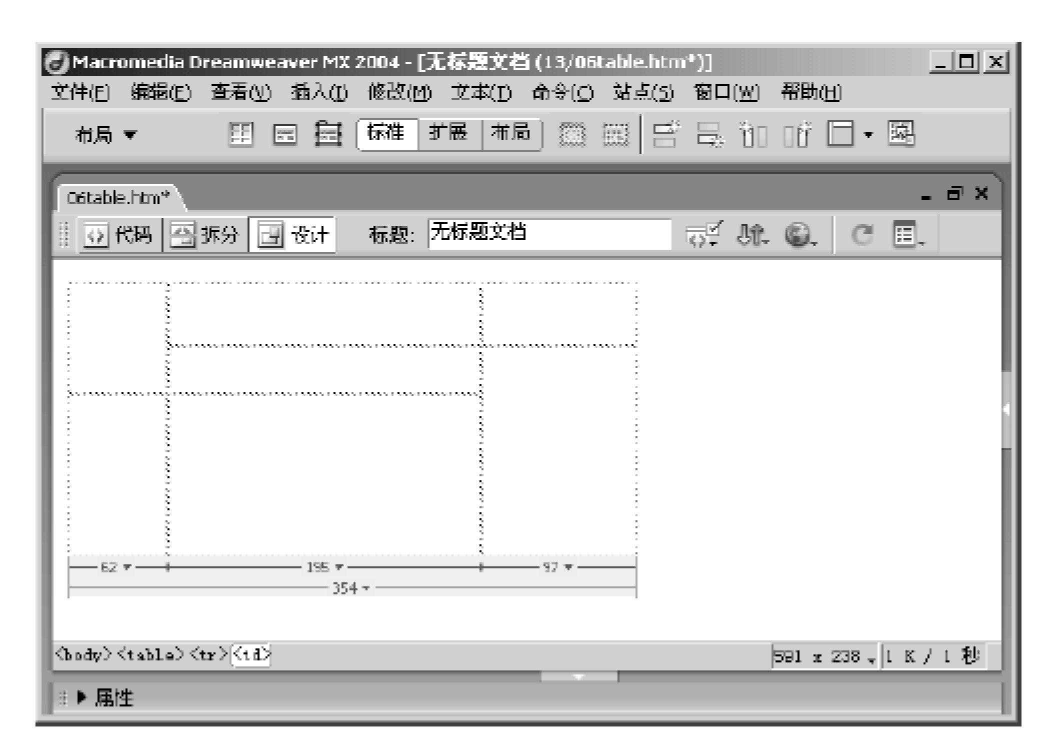


图 13-25　退出布局模式

13.2.2　设置框架

框架是页面布局常见的一种方式，在 Dreamweaver 中也可以很容易地实现，具体操作步骤如下：

（1）启动 Dreamweaver 后选择“文件”|“新建”命令，在打开的“新建文档”对话框中设置框架集结构，如图 13-26 所示。

（2）选择相应的框架集结构后，单击“创建”按钮可以创建一个框架页面。选择“文件”|“保存”命令可以保存框架集页面，此时框架窗口页面为空白页。

（3）选中框架可以在“属性”面板中设置框架的边框和边框颜色等，如图 13-27 所示。



图 13-26　创建框架页面　　　　图 13-27　设置框架的边框属性

（4）单击“布局”工具栏中的“框架”按钮，可以在页面中插入新的框架结构。

（5）将光标移动到框架的边框上，拖动框架边框，可以调整框架的布局，如图 13-28 所示。

（6）在文档窗口中按住 Alt 键，然后用鼠标单击一个框架，选中该框架。在“属性”面板中可以设置该框架窗口的属性，如图 13-29 所示。

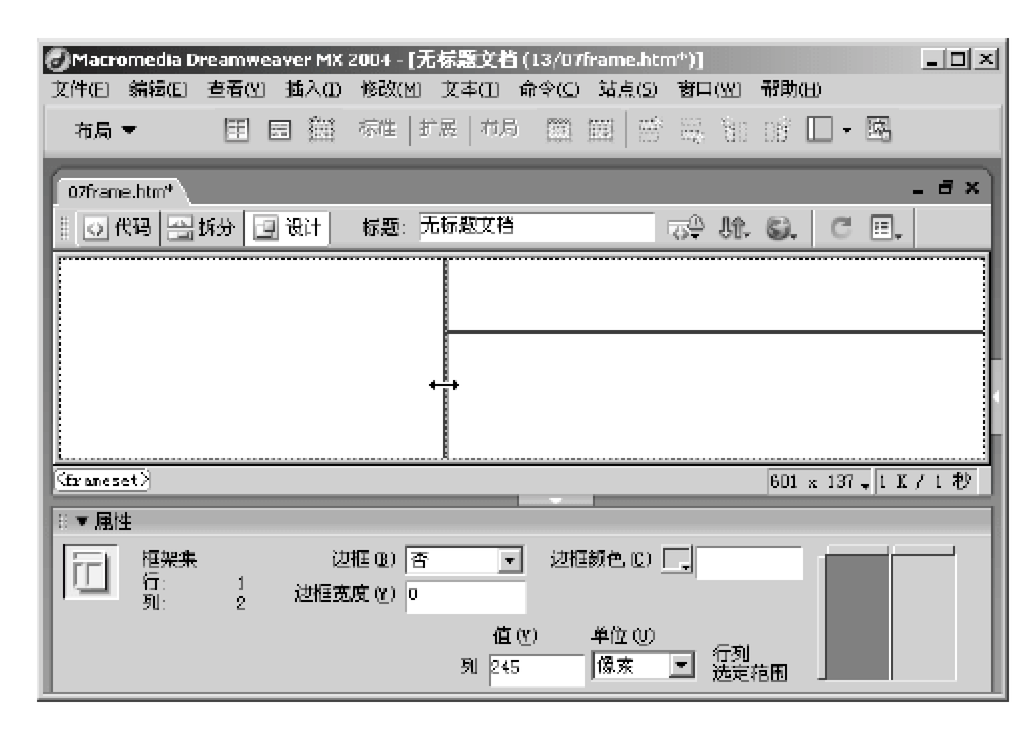




图 13-28　调整框架　　　　图 13-29　设置框架窗口的属性

（7）如果要完整保存框架结构，就要保存框架集文件和所有的框架窗口文件。

13.2.3　使用层进行布局

层是 Dreamweaver 中有代表性的对象之一，在制作网页时经常用到，其在元素的定位方面有着不可替代的作用和无可比拟的优势。使用层进行布局的方法如下：

（1）打开一个 HTML 文件，如图 13-30 所示。

（2）单击“布局”工具栏中的“描绘层”按钮，拖动鼠标左键在页面中绘制一个层，如图 13-31 所示。

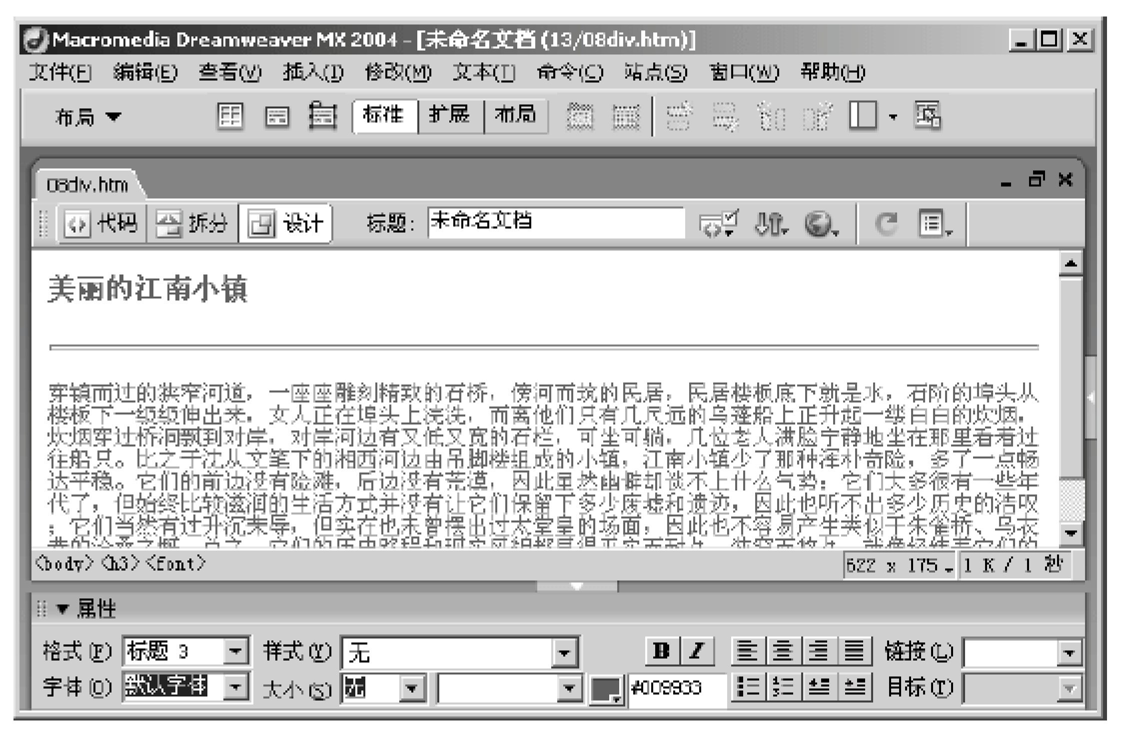


图 13-30　打开要插入层的页面



图 13-31　插入一个层

（3）在层内可以直接添加内容，包括文字、图像、表格等。

（4）选中这个层，在“属性”面板中可以设置层的位置、Z 轴（即层叠顺序）值、层的背景等，此处设置如图 13-32 所示。

（5）如果为一个页面添加多个层，那么 Z 轴的值越大，显示就越靠前，如图 13-33 所示。



图 13-32　设置层的属性



图 13-33　多个层的重叠效果

（6）选中层后，将鼠标移动到层的边线上，光标变为✥，此时可以拖动层，调整其位置。如果光标变为双箭头形状，则可以改变层的大小。

13.3　表单的使用

使用 Dreamweaver 可以在页面中随意插入各种表单。下面以最常用的几种表单为例进行详细介绍。

13.3.1　插入空白表单

（1）新建一个 HTML 文件，为文件命名后保存文件。

（2）单击工具栏中的分类列表，在下拉列表中选择“表单”，在工具栏中将显示表单类的按钮，如图 13-34 所示。



图 13-34　“表单”工具栏

（3）如果要添加表单元素，则单击“表单”按钮即可在页面中创建一个空白表单。

（4）此时在“属性”面板中可以设置表单的处理程序、提交方式等。其中，“表单名称”下方的文本框用于设置表单名称；“动作”则是设置表单的处理程序，可以单击文本框后的“选择文件”按钮选择处理程序文件，也可以直接在文本框中添加处理程序的名称，如“mailto:abcd@163.com”，如图 13-35 所示。

（5）运行程序，可以看到由于表单是空白的，在页面中并不显示出来，如图 13-36 所示。



图 13-35　设置表单的属性

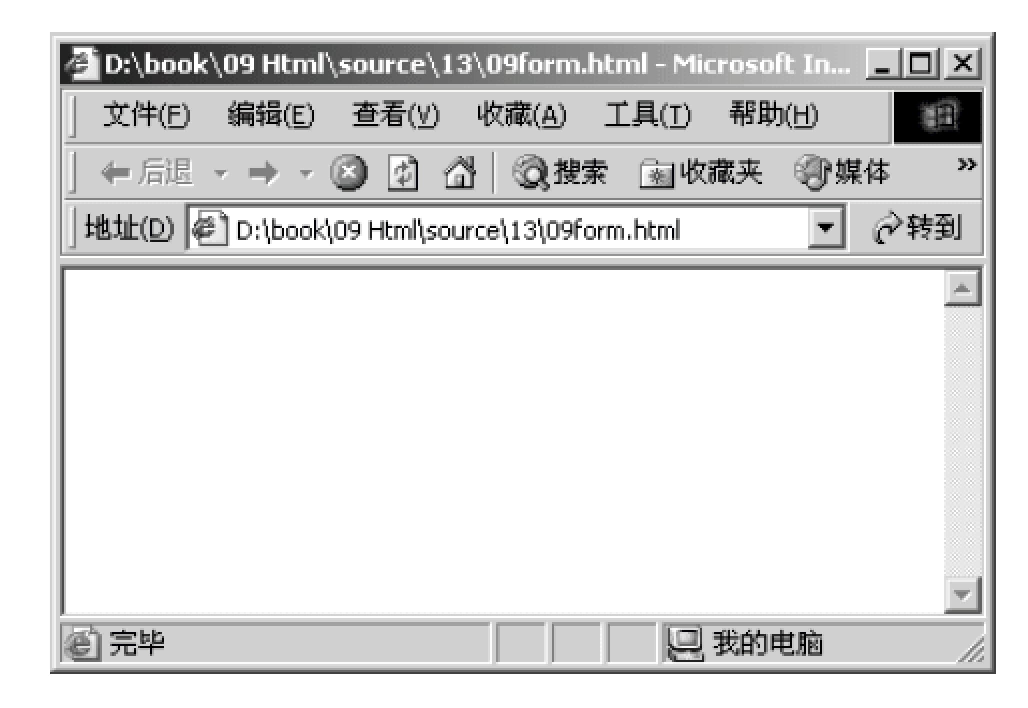


图 13-36　插入空白表单

13.3.2　插入文本类控件

在 Dreamweaver 中插入文本字段、文本域、密码域的方法比较类似，这里一起介绍。插入该类型控件的方法如下：

（1）新建一个 HTML 文件，为文件命名并保存。

（2）单击“表单”工具栏中的“文本字段控件”按钮，在页面中插入一个单行的文本域。此时在“属性”面板中可以设置表单的属性，如图 13-37 所示。

（3）单击选中页面中的文本框，此时可以在“属性”面板中设置该控件的名称、字符宽度（也就是文本框的长度）、最多字符数、初始值等内容，如图 13-38 所示。



图 13-37　设置表单属性

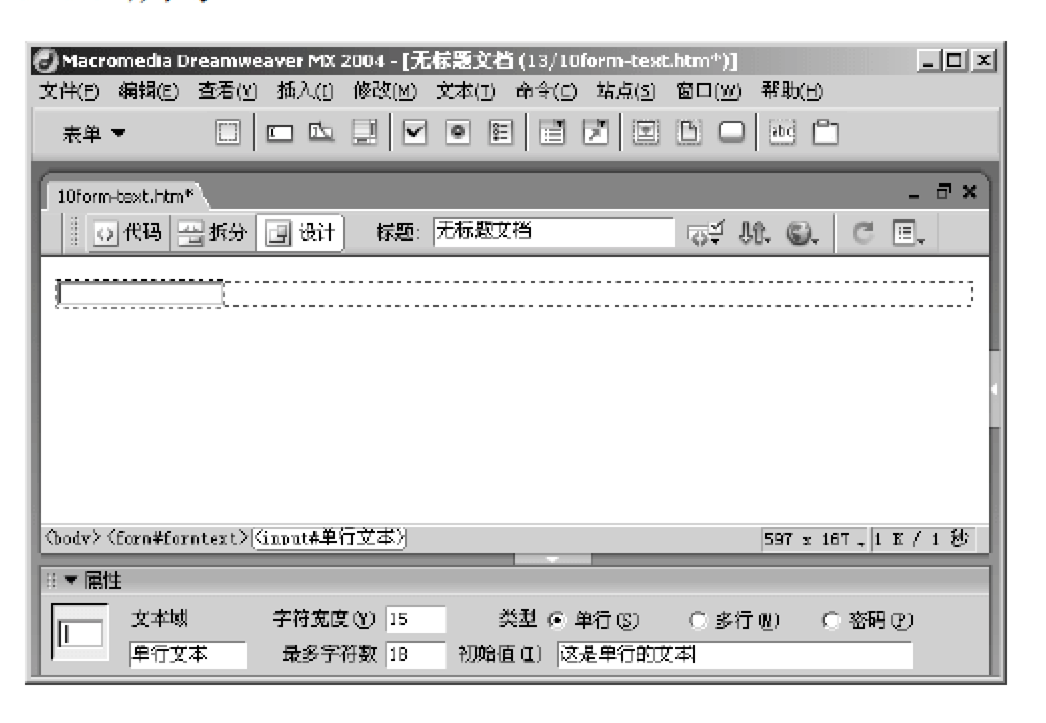


图 13-38　插入单行文本框

（4）单击“表单”工具栏中的“文本区域控件”按钮，可以在页面中添加一个多行的文本区域。选中该控件后也可以在“属性”面板中设置其各项参数，如图 13-39 所示。由图中可以看出，其实文本区域控件和文本字段控件可以实现相同的功能，只是默认情况下文本字段为单行，文本区域为多行，读者可以根据自己的喜好来选择。

（5）添加密码域的方法类似，一般情况下密码都采用单行的文本框。用前面介绍的方法添加一个文本字段后，在“属性”面板中选择“密码”类型即可完成密码域的添加。运行程序，可以看到添加的第三个文本域在输入内容时只显示星号“*”，如图 13-40 所示。

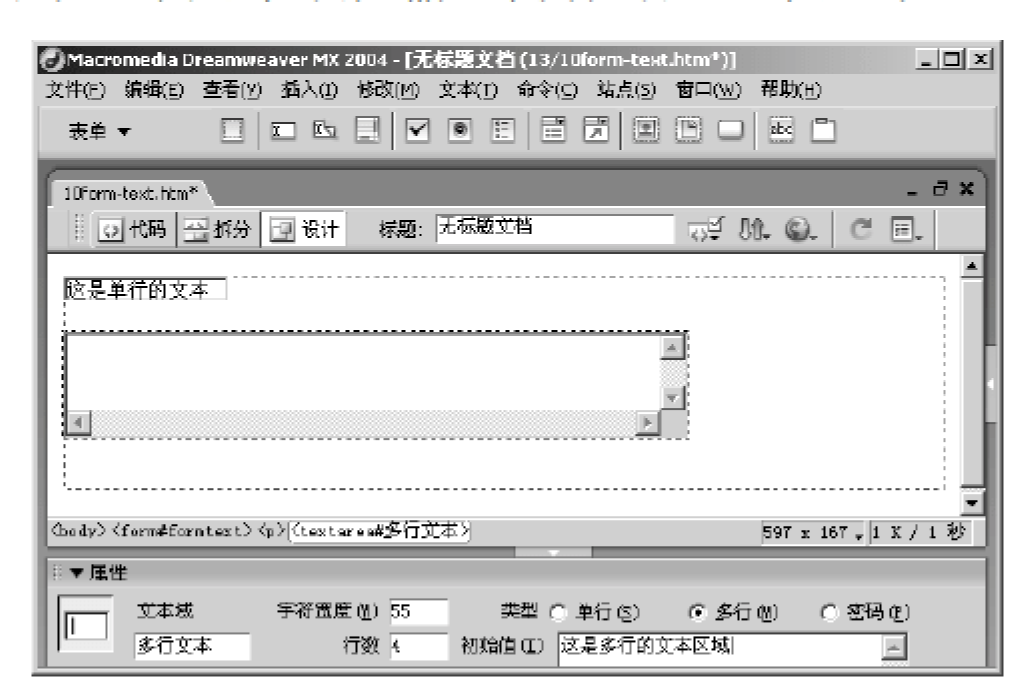


图 13-39　添加多行的文本区域



图 13-40　添加文本类表单的效果

13.3.3　插入选择区域

选择区域主要是指单选按钮、复选框。插入选择区域的方法如下：

（1）新建一个文件，为该文件命名并保存。

（2）单击“表单”工具栏中的“复选框”控件按钮，可以在页面上添加带有一个复选框的表单，此时可以直接在“属性”面板中设置表单的各种属性。

（3）选中复选框，在“属性”面板中可以设置该复选框的名称、值以及初始状态，如图 13-41 所示。

（4）由于复选框在显示时不能显示取值，因此一般在网页上都要添加该复选框的说明文字，如图 13-42 所示。



图 13-41　设置复选框的各种属性

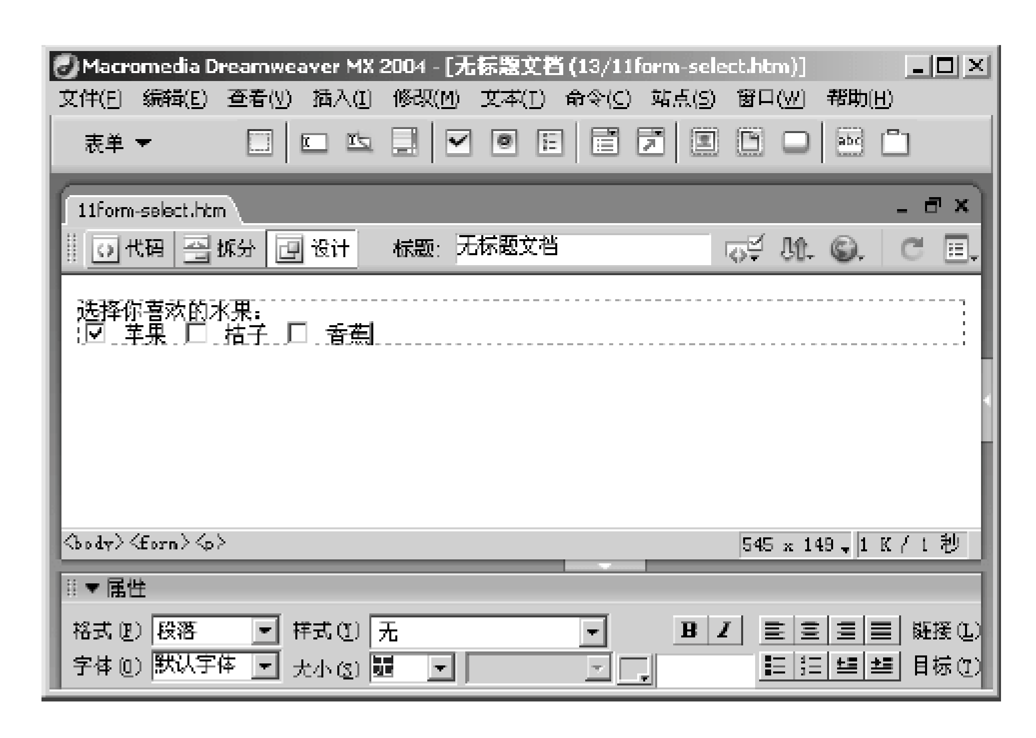


图 13-42　添加复选框的文字说明

（5）如果要添加单选按钮，单击工具栏中的“单选按钮”控件按钮，此时在页面上添加了一个单选按钮。选中该单选按钮可以在“属性”面板中设置该单选按钮的各种属性，如图 13-43 所示。需要注意的是，单选按钮同样需要添加文字说明。

（6）用同样的方法添加其他的单选按钮，值得注意的是，一个组中的单选按钮要取相同的名称，否则会出现如图 13-44 所示的状况，此时就不能称为真正的单选按钮了。

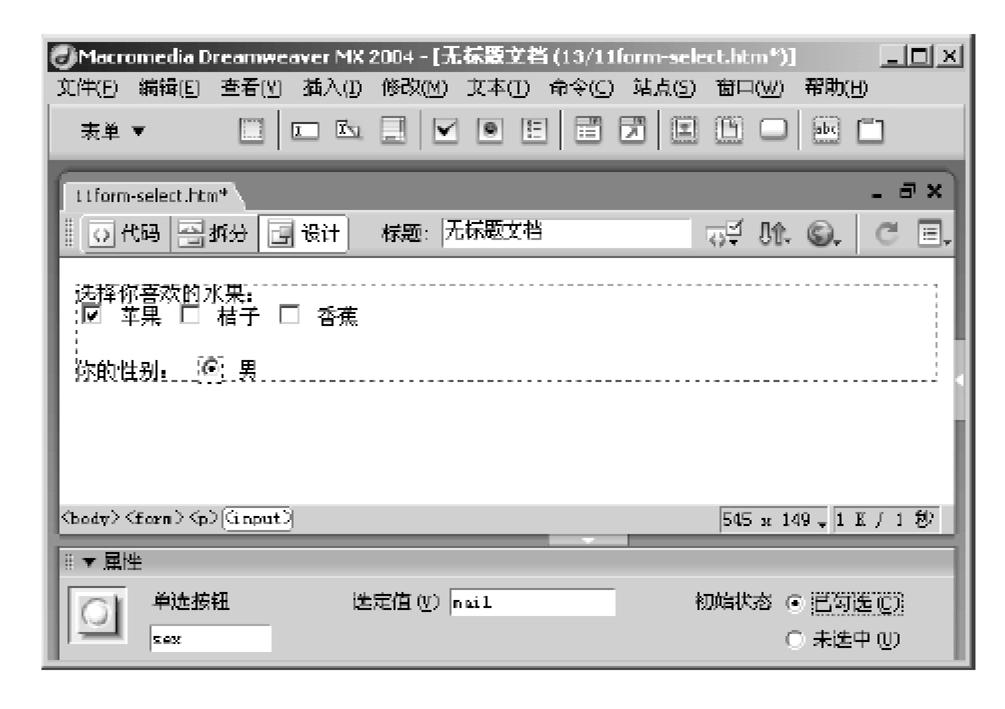


图 13-43　添加单选按钮



图 13-44　单选按钮设置了不同的名称效果

（7）添加单选按钮还可以单击“表单”工具栏中的“单选按钮组”控件按钮，此时弹出如图 13-45 所示的对话框。

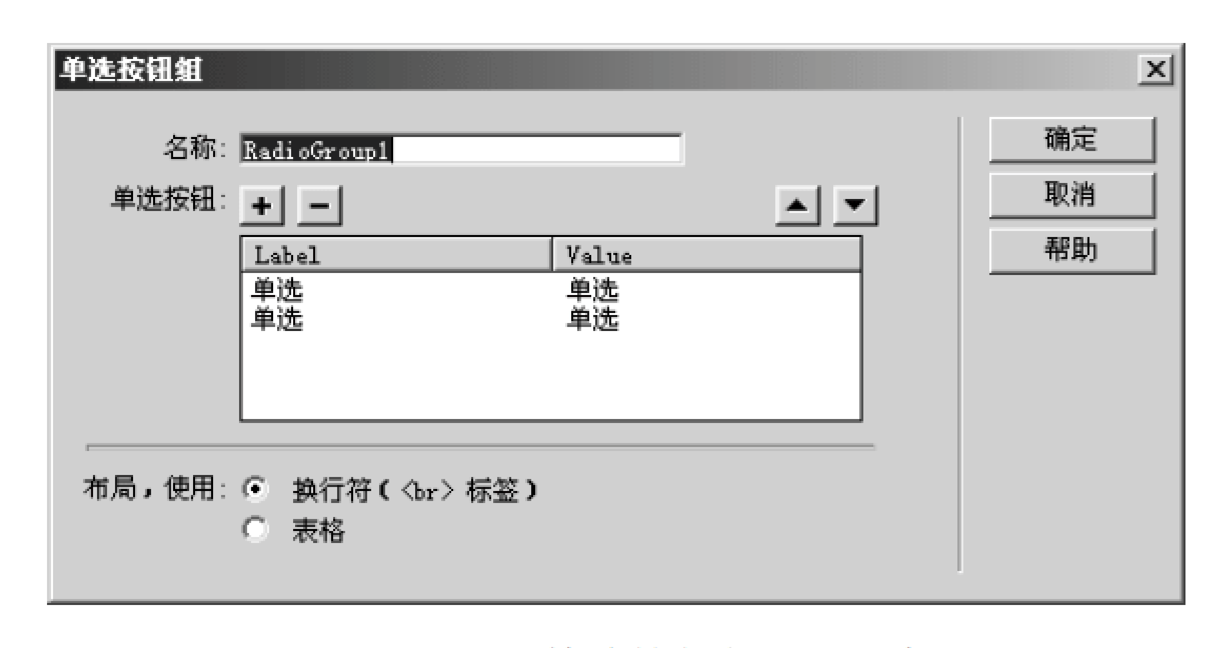


图 13-45　“单选按钮组”对话框

（8）在“名称”文本框中可以设置该单选按钮组（也就是该项选择）的名称，单击“添加”按钮可以在选择答案中增加一个单选按钮；单击“删除”按钮可以减少一个单选按钮。单击单选按钮的 Label 属性可以修改单选按钮在页面上显示的提示文字（即选择的内容）；单击单选按钮的 Value 属性可以修改单选按钮提交时的值，如图 13-46 所示。

（9）选择一个单选按钮的选项后，单击对话框中的“上移”按钮可以将该选项的位置提前；同样，如果单击“下移”按钮可以将该选项的位置向下移动。

（10）完成设置后单击“确定”按钮可以完成单选按钮组的添加。此时如果选中其中任何一个单选按钮，会发现它们的名称都是刚才设定的名称，如图 13-47 所示。



图 13-46　设置单选按钮组

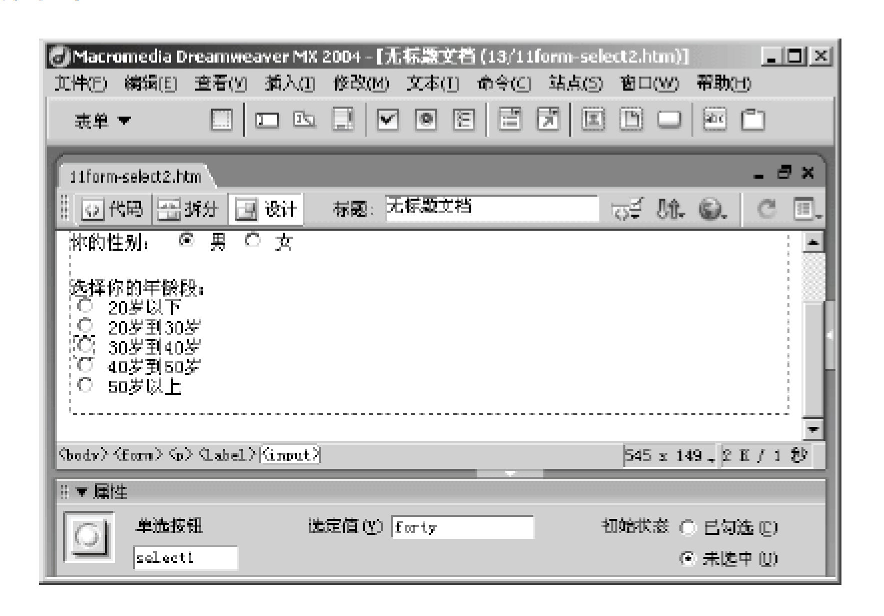


图 13-47　查看单选按钮的属性

13.3.4　插入菜单和列表

列表和菜单是比较节省空间的选择区域，在 Dreamweaver 中的添加方法如下：

（1）新建一个 HTML 文件，命名后保存该文件。

（2）单击“表单”工具栏中的“列表/菜单”控件按钮，在页面上添加一个菜单控件。选中该控件可以在“属性”面板中设置菜单的名称，或将控件设置为列表。

（3）如果要设置菜单的选项值，可以单击“属性”面板中的“列表值”按钮，打开“列表值”对话框，如图 13-48 所示。

（4）单击对话框中的“添加项目”按钮可以为菜单添加一个选项。在“项目标签”文本框中设置页面显示的文字，在“值”文本框中设置提交表单的值，如图 13-49 所示。单击对话框中的“上移”、“下移”按钮可以调整选项的排列顺序。完成后单击“确定”按钮完成菜单项目的添加。



图 13-48　“列表值”对话框

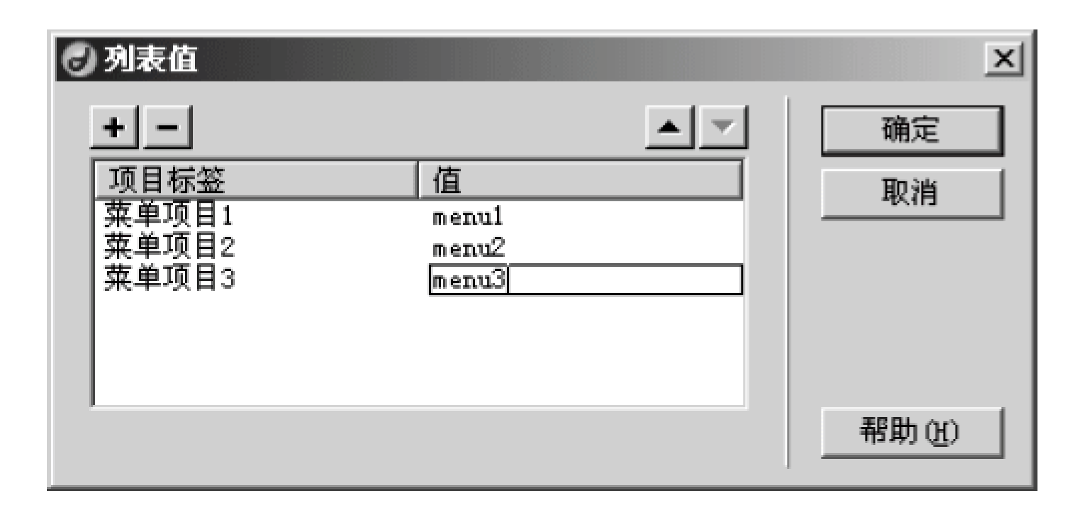


图 13-49　添加菜单的选项

（5）如果要添加列表，则选择控件后在“属性”面板中将控件类型更改为“列表”，此时可以设

置列表的高度（即默认显示的列表项）以及是否可以多选，如图 13-50 所示。

（6）单击“列表值”按钮，在打开的“列表值”对话框中同样可以添加列表的各选项值。完成后保存页面，运行程序可以看到效果如图 13-51 所示。



图 13-50　设置列表的属性



图 13-51　添加菜单和列表的效果

13.3.5　添加跳转菜单

跳转菜单是一种比较特殊的菜单，它通过菜单设置页面的链接。实现方法如下：

（1）创建一个 HTML 文件，为该文件命名后保存。

（2）单击页面中的“跳转菜单”控件按钮，打开如图 13-52 所示的“插入跳转菜单”对话框。

（3）在该对话框的“菜单项”中可以显示设置的菜单项目，在“文本”对话框设置菜单项的显示内容；在“选择时，转到 URL“文本框中可以设置跳转的目标页面。单击“添加项目”按钮可以增加新的菜单项；单击“删除”按钮可以删除当前的菜单项。在对话框的下半部分可以设置跳转菜单的名称和附加选项，如图 13-53 所示。

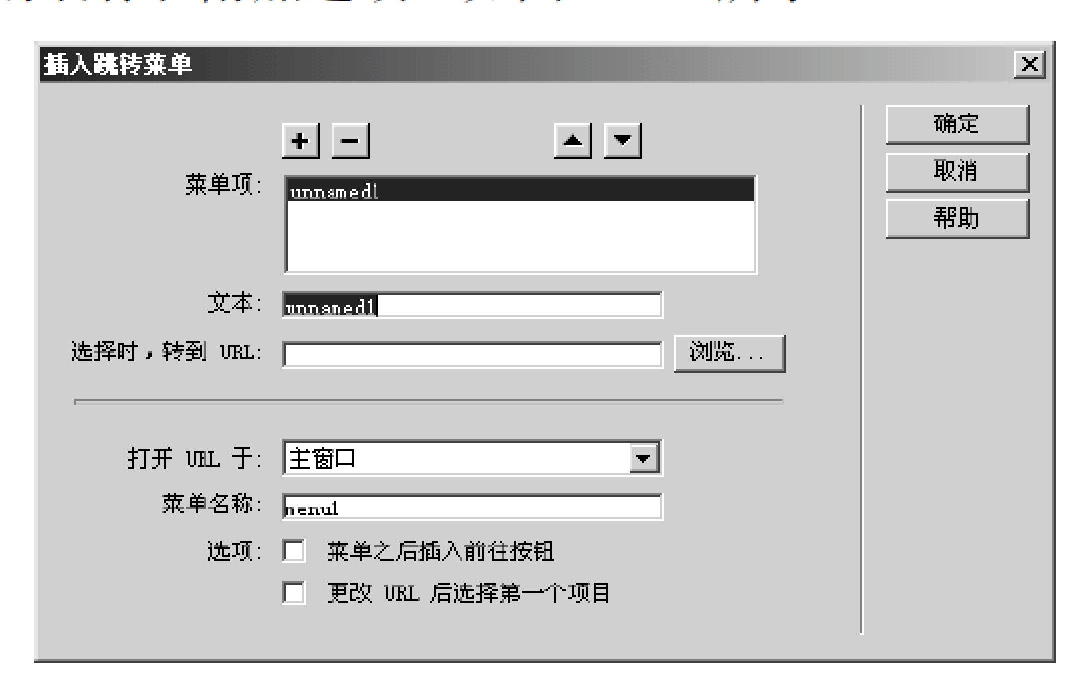


图 13-52　“插入跳转菜单”对话框



图 13-53　设置跳转菜单的属性

注意： 添加的跳转菜单项在运行时，只有更改了选项才能够进行成功跳转，因此最好加上一个没有跳转效果的默认选项。

（4）设置完成后，单击“确定”按钮在页面中实现跳转菜单的添加。运行程序可以看到如图 13-54 所示的效果。选择菜单中的“跳转项目 1”，页面会跳转到设置的 exam.html 文件中，如图 13-55 所示。




图 13-54　跳转菜单的效果

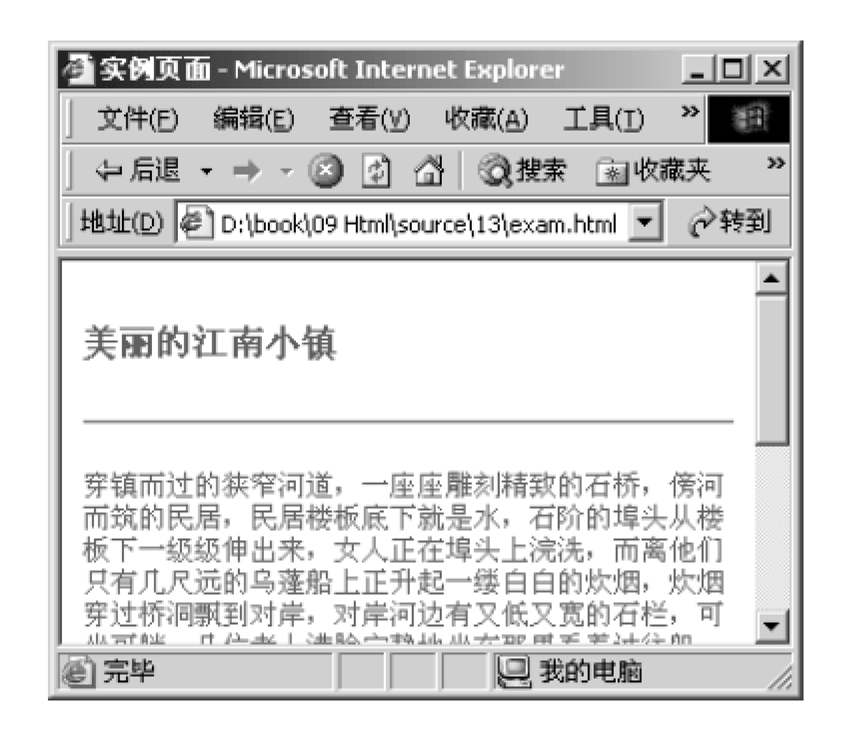


图 13-55　跳转后的页面

提示：如果在页面中选中该菜单，并在“属性”面板中将其类型更改为“列表”，那么最后实现的就是跳转列表的效果。

13.3.6　添加按钮

页面中还有一种常见的表单元素，即按钮。按钮主要分为提交按钮、重置按钮和普通按钮，而这 3 种按钮的添加方法基本相同。

（1）新建一个文件，为该文件命名后保存。

（2）单击“表单”工具栏中的“按钮”控件，可以在页面中添加一个按钮。默认情况下同时添加一个表单，此时可以在“属性”面板中设置表单的各项属性。

（3）选中该按钮，可以在“属性”面板中设置按钮控件的名称、标签（即按钮上的文字）、动作（即按钮类型）等。动作主要包括“提交表单”（提交按钮）、“无”（普通按钮）和“重设表单”（重置按钮），如图 13-56 所示。



图 13-56　按钮控件的各项参数

13.4　添加各种类型的文本

在 Dreamweaver 中可以添加各种类型的文字，包括标题文字、列表项、粗体文字等。添加文本的方法如下：

（1）新建一个 HTML 文件，为文件命名后保存文件。

（2）单击页面上工具栏的分类列表，在下拉列表中选择“文本”，在工具栏中将显示文本类的按钮，如图 13-57 所示。



图 13-57　“文本”工具栏

（3）选择相应类型的文本按钮（如 h1），在页面中输入该类型的文本内容。双击选中文本内容后在“属性”面板中可以设置该类文字的样式，包括字体、字号、颜色等，如图 13-58 所示。

（4）完成后按下 Ctrl+Enter 组合键即可跳出该类文本，再按 Enter 键则进入到下一段落的文本编辑中。如果要添加其他类型的文本，只需在单击相应的文本类型后输入文字即可。

（5）此实例中添加标题文本、编号列表、段落文字，并设置不同的文本样式，效果如图 13-59 所示。

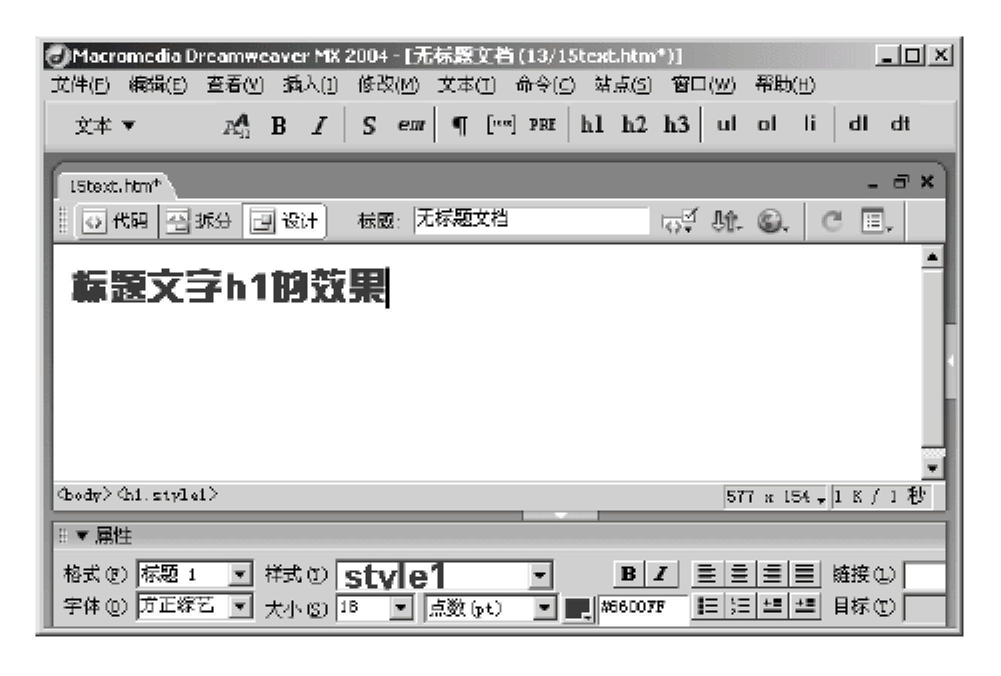


图 13-58　添加 h1 类型的文本

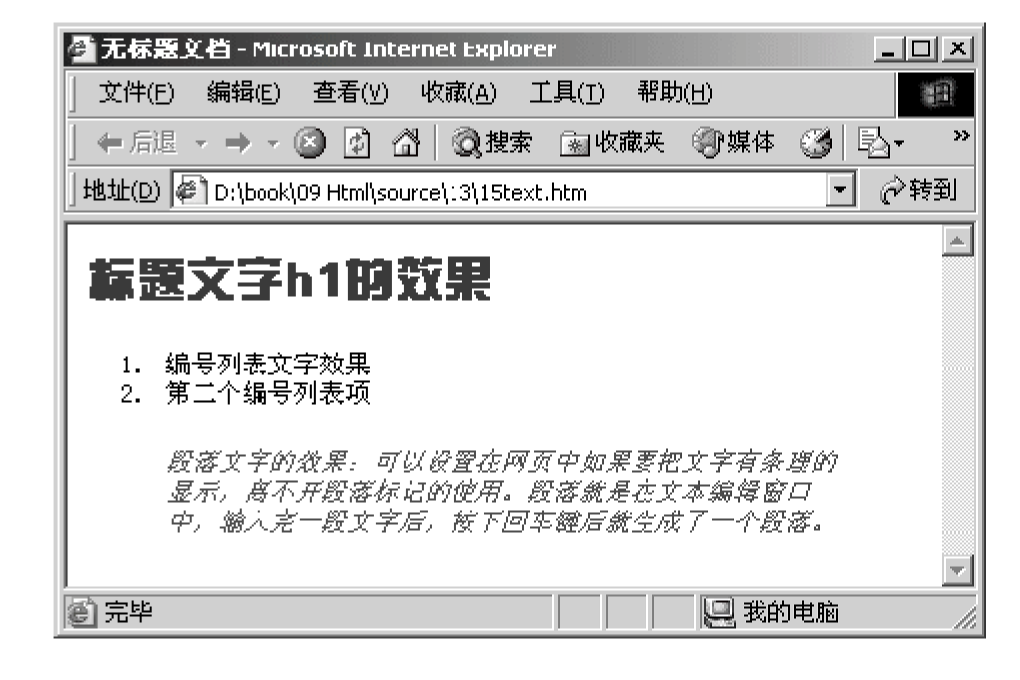


图 13-59　在 Dreamweaver 中添加各种类型的文本

13.5　添加 HTML 网页的辅助内容

HTML 网页的辅助内容主要指水平线、文件头、设置脚本属性等。

13.5.1　添加水平线

在 HTML 页面中添加水平线的具体步骤如下：

（1）新建一个文件，在页面中添加部分内容后保存，如图 13-60 所示。

（2）将光标移动到要添加水平线的位置，选择“插入”|HTML|“水平线”命令，或者在工具栏中

显示的 HTML 类中单击“水平线”按钮。此时在页面中添加了一条默认属性的水平线，且水平线处于选中状态。

（3）在“属性”面板中可以更改水平线的大小、对齐方式等，此处设置的效果如图 13-61 所示。

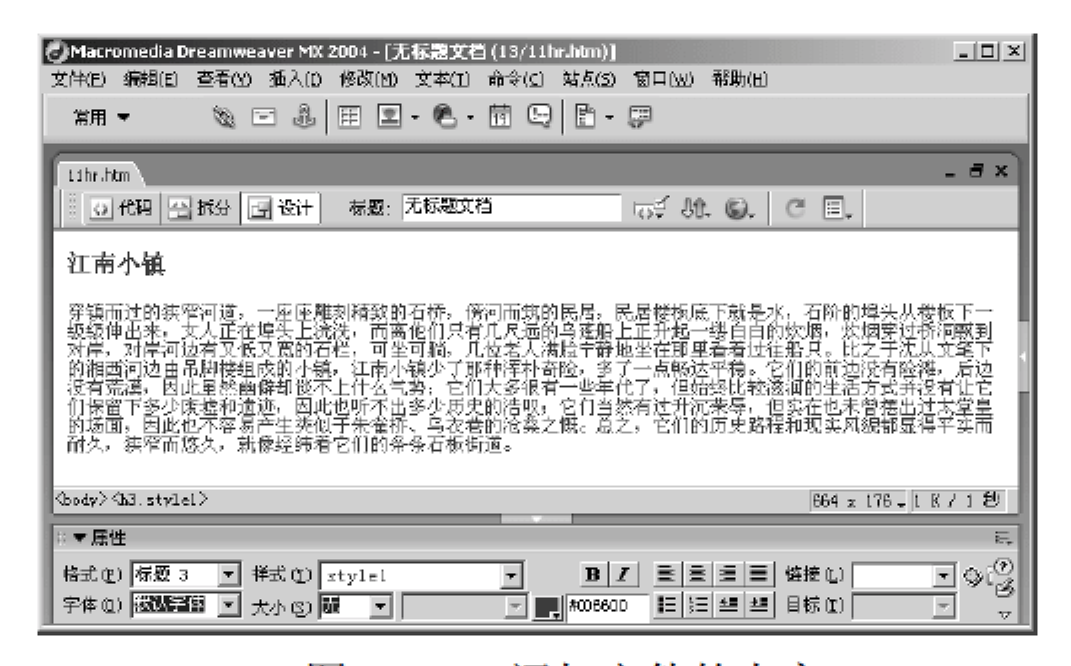


图 13-60　添加文件的内容

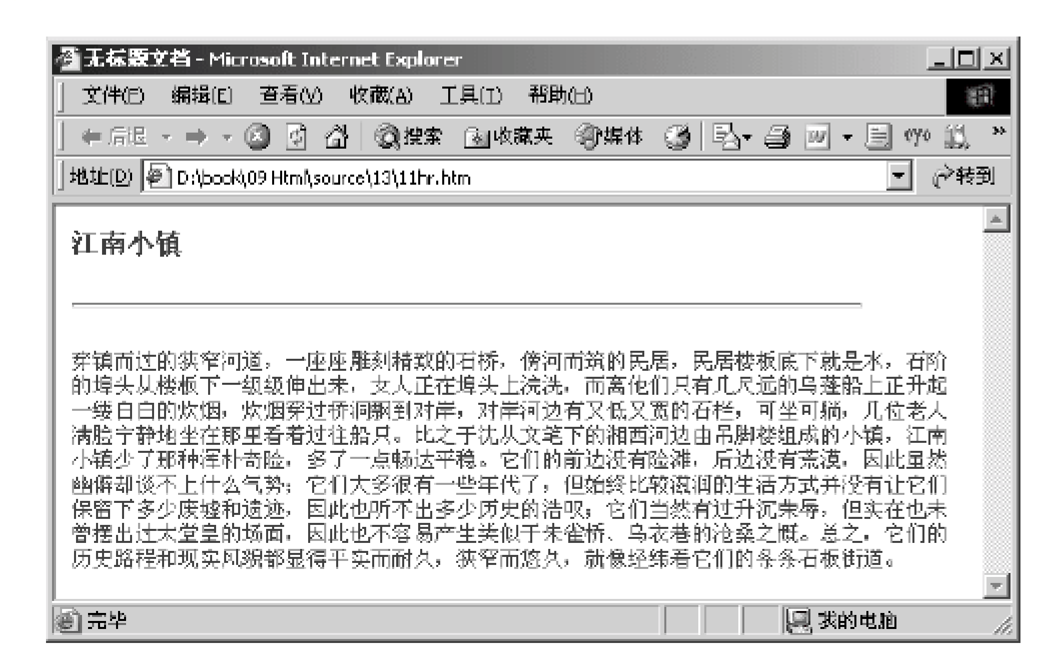


图 13-61　在页面中添加水平线

13.5.2　设置文件元信息

文件的元信息主要是用来定义页面信息的名称、关键字、作者等内容，并不显示在页面中。在 Dreamweaver 中添加元信息的方法快捷简单。

（1）新建一个文件，将该文件命名为 12meta.htm 并保存文件。

（2）选择“插入”|HTML|“文件头标签”|META 命令，打开如图 13-62 所示的对话框。

（3）在该对话框中的“属性”下拉列表框中可以设置 Meta 属性类型，在“值”文本框中设置具体的属性名，在“内容”文本框中则是具体属性的取值，如图 13-63 所示。此处设置了作者名称为“张王李赵”。

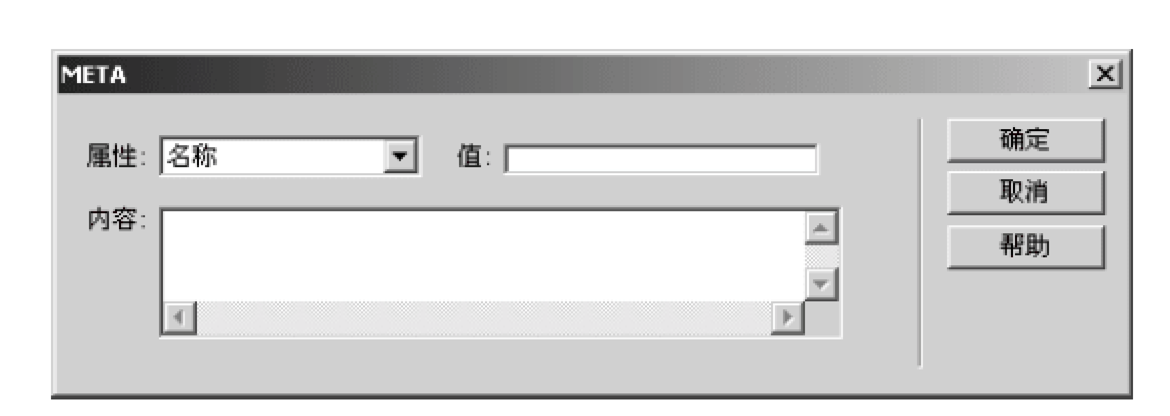


图 13-62　META 对话框



图 13-63　设置 META 属性

（4）完成后单击“确定”按钮则可以完成该属性的设置。

（5）如果要设置页面的关键字，可以选择“插入”|HTML|“文件头标签”|“关键字”命令，在弹出的“关键字”对话框中设置页面的关键字，各个关键字之间用英文逗号“,”隔开，如图 13-64 所示。完成后单击“确定”按钮在页面中添加该属性的设置。

（6）选择“插入”|HTML|“文件头标签”|“刷新”命令，在弹出的“刷新”对话框中设置页面的刷新或跳转时间。如果设置刷新，则在“操作”单选按钮组中选中“刷新此文档”；如果要设置跳转，则选中“转到 URL”单选按钮并在后面的文本框中设置要跳转的页面，如图 13-65 所示，设置了页面在 3 秒之后跳转到 exam.html 文件窗口。

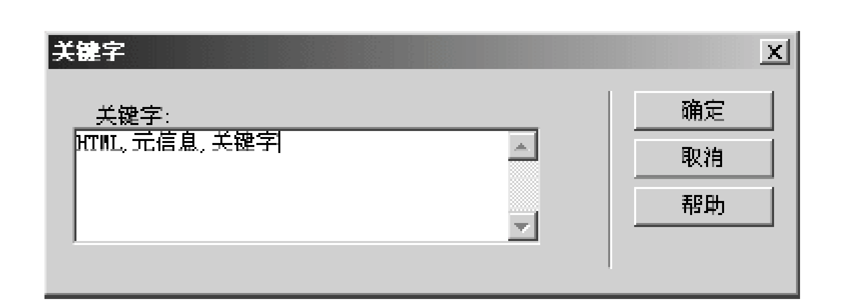




图 13-64　设置页面关键字



图 13-65　设置页面跳转

（7）在页面中添加其他元信息的方法与此类似，这里就不再重复说明，读者可以通过练习逐渐领会。

13.5.3　添加脚本

使用 Dreamweaver 在页面中添加脚本语言代码的方法如下：

（1）打开一个 HTML 文件。

（2）选择“插入”|HTML|“脚本对象”|“脚本”命令，可以打开“脚本”对话框。在对话框的“语言”下拉列表框中可以设置脚本语言，如图 13-66 所示。

（3）在“内容”文本域中直接添加脚本程序。此处添加脚本程序如下：

alert("使用脚本语言弹出一个窗口");

（4）单击“确定”按钮，出现如图 13-67 所示的提示窗口。

（5）单击“确定”按钮完成脚本的添加。如果希望不再显示这个信息，选中窗口中的复选框即可。运行程序的效果如图 13-68 所示。

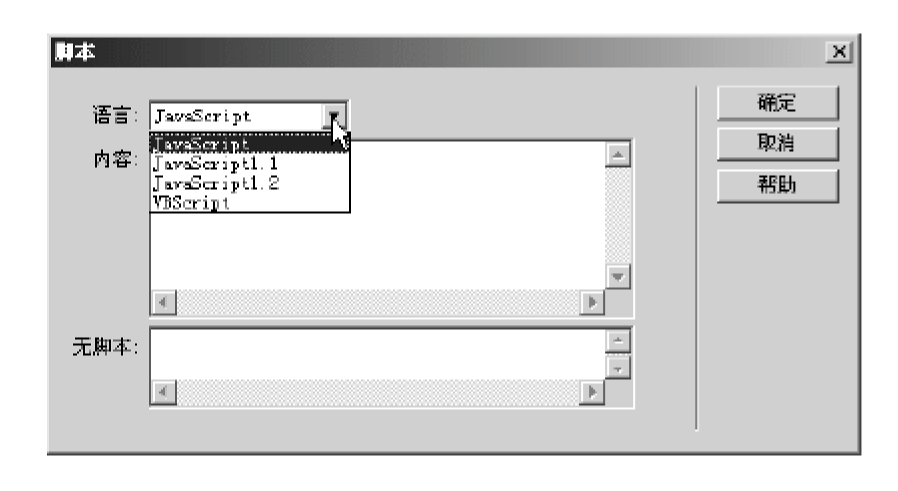


图 13-66　设置脚本语言



图 13-67　不可见元素提示窗口

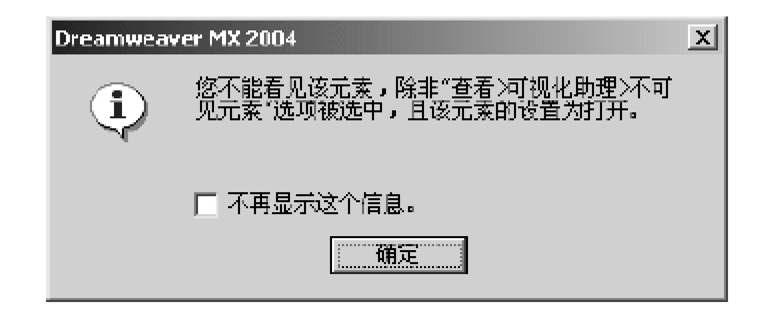


图 13-68　添加脚本后的运行效果

在 Dreamweaver 中还提供了其他一些功能，读者在学习完本书之后可以参考有关 Dreamweaver 的书籍加以了解，从而创建更美观、功能更完善的网页程序。

附录　颜色关键字对照表

颜 色 名 称	中 文 名 称	十六进制RGB	十进制RGB
aliceblue	艾利斯兰	#f0f8ff	240,248,255
antiquewhite	古董白	#faebd7	250,235,215
aqua	浅绿色	#00ffff	0,255,255
aquamarine	碧绿色	#7fffd4	127,255,212
azure	天蓝色	#f0ffff	240,255,255
beige	米色	#f5f5dc	245,245,220
bisque	桔黄色	#ffe4c4	255,228,196
black	黑色	#000000	0,0,0
blanchedalmond	白杏色	#ffebcd	255,235,205
blue	蓝色	#0000ff	0,0,255
blueviolet	紫罗兰色	#8a2be2	138,43,226
brown	褐色	#a52a2a	165,42,42
burlywood	实木色	#deb887	222,184,135
cadetblue	军蓝色	#5f9ea0	95,158,160
chartreuse	黄绿色	#7fff00	127,255,0
chocolate	巧克力色	#d2691e	210,105,30
coral	珊瑚色	#ff7f50	255,127,80
cornflowerblue	菊蓝色	#6495ed	100,149,237
cornsilk	米绸色	#fff8dc	255,248,220
crimson	暗深红色	#dc143c	220,20,60
cyan	青色	#00ffff	0,255,255
darkblue	暗蓝色	#00008b	0,0,139
darkcyan	暗青色	#008b8b	0,139,139
darkgoldenrod	暗金黄色	#b8860b	184,134,11
darkgray	暗灰色	#a9a9a9	169,169,169
darkgreen	暗绿色	#006400	0,100,0
darkkhaki	暗黄褐色	#bdb76b	189,183,107
darkmagenta	暗洋红	#8b008b	139,0,139
darkolivegreen	暗橄榄绿	#556b2f	85,107,47
darkorange	暗桔黄色	#ff8c00	255,140,0
darkorchid	暗紫色	#9932cc	153,50,204
darkred	暗红色	#8b0000	139,0,0
darksalmon	暗肉色	#e9967a	233,150,122
darkseagreen	暗海蓝色	#8fbc8f	143,188,143

续表

颜 色 名 称	中 文 名 称	十六进制RGB	十进制RGB
darkslateblue	暗灰蓝色	#483d8b	72,61,139
darkslategrey	暗瓦灰色	#2f4f4f	47,79,79
darkturquoise	暗宝石绿	#00ced1	0,206,209
darkviolet	暗紫罗兰色	#9400d3	148,0,211
deeppink	深粉红色	#ff1493	255,20,147
deepskyblue	深天蓝色	#00bfff	0,191,255
dimgray	暗灰色	#696969	105,105,105
dodgerblue	闪蓝色	#1e90ff	30,144,255
firebrick	火砖色	#b22222	178,34,34
floralwhite	花白色	#fffaf0	255,250,240
forestgreen	森林绿	#228b22	34,139,34
fuchsia	紫红色	#ff00ff	255,0,255
gainsboro	淡灰色	#dcdcdc	220,220,220
ghostwhite	幽灵白	#f8f8ff	248,248,255
gold	金色	#ffd700	255,215,0
goldenrod	金麒麟色	#daa520	218,165,32
gray	灰色	#808080	128,128,128
green	绿色	#008000	0,128,0
greenyellow	黄绿色	#adff2f	173,255,47
honeydew	蜜色	#f0fff0	240,255,240
hotpink	热粉红色	#ff69b4	255,105,180
indianred	印第安红	#cd5c5c	205,92,92
indigo	靛青色	#4b0082	75,0,130
ivory	象牙色	#fffff0	255,255,240
khaki	黄褐色	#f0e68c	240,230,140
lavender	淡紫色	#e6e6fa	230,230,250
lavenderblush	淡紫红	#fff0f5	255,240,245
lawngreen	草绿色	#7cfc00	124,252,0
lemonchiffon	柠檬绸色	#fffacd	255,250,205
lightblue	亮蓝色	#add8e6	173,216,230
lightcoral	亮珊瑚色	#f08080	240,128,128
lightcyan	亮青色	#e0ffff	224,255,255
lightgoldenrodyellow	亮金黄色	#fafad2	250,250,210
lightgray	亮灰色	#d3d3d3	211,211,211
lightgreen	亮绿色	#90ee90	144,238,144
lightpink	亮粉红色	#ffb6c1	255,182,193
lightsalmon	亮肉色	#ffa07a	255,160,122
lightseagreen	亮海蓝色	#20b2aa	32,178,170
lightskyblue	亮天蓝色	#87cefa	135,206,250

续表

颜色名称	中文名称	十六进制RGB	十进制RGB
lightslategray	亮蓝灰	#778899	119,136,153
lightsteelblue	亮钢蓝色	#b0c4de	176,196,222
lightyellow	亮黄色	#ffffe0	255,255,224
lime	酸橙色	#00ff00	0,255,0
limegreen	橙绿色	#32cd32	50,205,50
linen	亚麻色	#faf0e6	250,240,230
magenta	红紫色	#ff00ff	255,0,255
maroon	粟色	#800000	128,0,0
mediumaquamarine	中绿色	#66cdaa	102,205,170
mediumblue	中蓝色	#0000cd	0,0,205
mediumorchid	中粉紫色	#ba55d3	186,85,211
mediumpurple	中紫色	#9370db	147,112,219
mediumseagreen	中海蓝	#3cb371	60,179,113
mediumslateblue	中暗蓝色	#7b68ee	123,104,238
mediumspringgreen	中春绿色	#00fa9a	0,250,154
mediumturquoise	中绿宝石	#48d1cc	72,209,204
mediumvioletred	中紫罗蓝色	#c71585	199,21,133
midnightblue	中灰蓝色	#191970	25,25,112
mintcream	薄荷色	#f5fffa	245,255,250
mistyrose	浅玫瑰色	#ffe4e1	255,228,225
moccasin	鹿皮色	#ffe4b5	255,228,181
navajowhite	纳瓦白	#ffdead	255,222,173
navy	海军色	#000080	0,0,128
oldlace	老花色	#fdf5e6	253,245,230
olive	橄榄色	#808000	128,128,0
olivedrab	深绿褐色	#6b8e23	107,142,35
orange	橙色	#ffa500	255,165,0
orangered	红橙色	#ff4500	255,69,0
orchid	淡紫色	#da70d6	218,112,214
palegoldenrod	苍麒麟色	#eee8aa	238,232,170
palegreen	苍绿色	#98fb98	152,251,152
paleturquoise	苍宝石绿	#afeeee	175,238,238
palevioletred	苍紫罗蓝色	#db7093	219,112,147
papayawhip	番木色	#ffefd5	255,239,213
peachpuff	桃色	#ffdab9	255,218,185
peru	秘鲁色	#cd853f	205,133,63
pink	粉红色	#ffc0cb	255,192,203
plum	洋李色	#dda0dd	221,160,221
powderblue	粉蓝色	#b0e0e6	176,224,230

续表

颜 色 名 称	中 文 名 称	十六进制RGB	十进制RGB
purple	紫色	#800080	128,0,128
red	红色	#ff0000	255,0,0
rosybrown	褐玫瑰红	#bc8f8f	188,143,143
royalblue	皇家蓝	#4169e1	65,105,225
saddlebrown	重褐色	#8b4513	139,69,19
salmon	鲜肉色	#fa8072	250,128,114
sandybrown	沙褐色	#f4a460	244,164,96
seagreen	海绿色	#2e8b57	46,139,87
seashell	海贝色	#fff5ee	255,245,238
sienna	赭色	#a0522d	160,82,45
silver	银色	#c0c0c0	192,192,192
skyblue	天蓝色	#87ceeb	135,206,235
slateblue	石蓝色	#6a5acd	106,90,205
slategray	灰石色	#708090	112,128,144
snow	雪白色	#fffafa	255,250,250
springgreen	春绿色	#00ff7f	0,255,127
steelblue	钢蓝色	#4682b4	70,130,180
tan	茶色	#d2b48c	210,180,140
teal	水鸭色	#008080	0,128,128
thistle	蓟色	#d8bfd8	216,191,216
tomato	西红柿色	#ff6347	255,99,71
turquoise	青绿色	#40e0d0	64,224,208
violet	紫罗蓝色	#ee82ee	238,130,238
wheat	浅黄色	#f5deb3	245,222,179
white	白色	#ffffff	255,255,255
whitesmoke	烟白色	#f5f5f5	245,245,245
yellow	黄色	#ffff00	255,255,0
yellowgreen	黄绿色	#9acd32	154,205,50

读者意见反馈卡

您购买的书名：________ 您的姓名：________ 性 别：☐男 ☐女
年龄：________ 文化程度：________ 职 业：________
邮编：________ 通信地址：________ E-mail：________
您常用的软件：1 ________ 2 ________ 3 ________ 4 ________

您购买本书的原因（可多选）：

☐封面与装帧 ☐引言目录 ☐正文内容 ☐丛书风格 ☐价格 ☐光盘 ☐专业性强 ☐别人介绍
☐出版社或作者名声 ☐售后服务

本书最令您满意的是（可多选）：

☐专业性强、覆盖面广 ☐内容翔实、定位准确 ☐精益求精、售后服务

您可以承受的图书价格：

☐20 元以下 ☐30 元以下 ☐40 元以下 ☐50 元以下 ☐只要内容好，不论价格

您对本书的评价：

封面装帧：☐很好 ☐较好 ☐一般 ☐不满意 建议________
印刷质量：☐很好 ☐较好 ☐一般 ☐不满意 建议________
正文质量：☐很好 ☐较好 ☐一般 ☐不满意 建议________
写作风格：☐很好 ☐较好 ☐一般 ☐不满意 建议________
专业水平：☐很好 ☐较好 ☐一般 ☐不满意 建议________

您希望增加哪些图书选题：1________ 2________ 3________

您认为本书有哪些错误：

章____节____页码____行____列____ 图号____ 错误________应改为________
章____节____页码____行____列____ 图号____ 错误________应改为________
章____节____页码____行____列____ 图号____ 错误________应改为________
章____节____页码____行____列____ 图号____ 错误________应改为________

您的其他建议：

1________
2________
3________

请填好本卡后寄给：

清华大学校内金地公司 邮编：100084 电话：（010）62791976-220
《HTML 网页设计参考手册》编辑部收
传真：（010）62788903 公司网址：www.thjd.com.cn E-mail：oyzx_sp@263.net
如需本书可与本编辑部联系邮购，汇款请按以上地址填写，另加邮费 15%（挂号）